

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет Інформатики та обчислювальної техніки
Кафедра Автоматики та управління в технічних системах

«До захисту допущено»

Завідувач кафедри

_____ Ролік О. І.
(підпис) (ініціали, прізвище)

«__» _____ 2019 р.

Магістерська дисертація

зі спеціальності 126 Інформаційні системи та технології

на тему: «Рекомендаційна система пошуку закладів харчування на
основі платформи обміну миттєвими повідомленнями»

Виконав: студент 6-го курсу, групи ІА-82мп
(шифр групи)

_____ Федорко Лізавета Антонівна
(прізвище, ім'я, по батькові) (підпис)

Науковий керівник доцент каф. АУТС, к.т.н. Букасов М.М.
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Рецензент _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____
(підпис)

Київ – 2019 р

**Національний технічний університет України
«Київський політехнічний інститут
імені Ігоря Сікорського»**

Факультет _____ інформатики та обчислювальної техніки
(повна назва)

Кафедра _____ автоматики та управління в технічних системах
(повна назва)

Рівень вищої освіти – другий (магістерський) за освітньо-професійною програмою

Спеціальність _____ 126 Інформаційні технології та системи
(код і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Ролік О. І.
(підпис) (ініціали, прізвище)

«__» _____ 2019 р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Федорко Лізаветі Антонівні

(прізвище, ім'я, по батькові)

1. Тема дисертації : _____ «Рекомендаційна система пошуку закладів харчування на основі платформи обміну миттєвими повідомленнями»

Науковий керівник дисертації _____ Букасов Максим Михайлович,
(прізвище, ім'я, по-батькові, науковий ступінь, вчене звання)
_____ к.т.н., доцент кафедри АУТС

затверджені наказом по університету від «__» _____ 2019 р. №__

2. Строк подання студентом дисертації _____

3. Об'єкт дослідження _____

4. Перелік завдання, які потрібно розробити: розробка системи пошуку закладів харчування на основі алгоритмів надання рекомендацій та платформи обміну миттєвими повідомленнями.

5. Орієнтовний перелік ілюстративного (графічного) матеріалу: діаграма варіантів використання, UML діаграма послідовності, UML діаграма розгортання системи для чат-боту, структурна схема система для системи надання рекомендацій, структурна

система, діаграма бази даних, UML діаграма діяльності для процесу пошуку закладів,
діаграма діяльності для процесу бронювання.

6. Консультанти розділів дисертації

7. Дата видачі завдання – 02.09.2019

Календарний план

№ з/п	Назва етапів виконання дипломного проекту	Строк виконання етапів проекту	Примітка
1.	Ознайомлення з завданням	02.09.2019	
2.	Огляд існуючих рішень та визначення вимог	15.10.2019	
3.	Вибір методів реалізації	1.11.2019	
4.	Вибір технологій	8.11.2019	
5.	Проектування архітектури системи	10.11.2019	
6.	Розроблення програмного забезпечення	25.11.2019	
7.	Тестування системи	27.11.2019	
8.	Оформлення документації	29.11.2019	
9.	Предзахист	3.12.2019	
10.	Представлення до захисту	19.12.2019	

Студент

Федорко Л.А.

(підпис)

(ініціали, прізвище)

Науковий керівник дисертації

Букасов М.М.

(підпис)

(ініціали, прізвище)

РЕФЕРАТ

Магістерська дисертація: 103 с., 35 рис., 66 табл., 9 додатків, 25 джерел.

Актуальність. У мешканців міста та туристів часто виникає потреба у пошуку закладів харчування. Інструментами для пошуку слугують пошукові системи та додатки, в яких досить важко знайти бажаний результат за декількома критеріями (місцезнаходження, цінова політика, конкретна позиція меню). Дані способи пошуку зазвичай націлені на розміщення реклами або обслуговують конкретні мережеві заклади харчування.

Ключовими умовами для вдалого підбору закладів харчування є різноманітність результатів, простота процесу пошуку та можливість здійснювати пошук на основі власних вподобань користувача. Платформи обміну миттєвими повідомленнями є швидким та поширеним способом комунікації між людьми. Але можливості таких платформ не обмежені даною функцією, вони можуть слугувати сервісом надання необхідної інформації у будь-якій сфері.

Метою є підвищення ефективності процесу підбору закладів харчування за рахунок автоматизації процесу пошуку закладів та бронювання.

Об'єктом дослідження є процес формування та надання пропозицій в закладах харчування.

Предметом є методи та моделі надання рекомендацій в закладах харчування.

Ключові слова: Рекомендаційна система, чат-бот, платформа обміну миттєвими повідомленнями, фільтрація вмісту, попередня обробка даних

ABSTRACT

Master's thesis: 113 pages, 35 figures, 66 tables, 25 sources

Relevance. Citizens and tourists often have problem of searching restaurants. These search methods are typically targeted at advertising. Search engines and applications are tools for it, but finding the desired result by several criteria is the main problem of it (location, pricing, specific menu item).

Instant messaging platforms are a fast and common way of communicating between people. But the capabilities of such platforms are not limited by this function. It can serve as a service to provide the necessary information in any field.

The goal is to increase the selection efficiency of restaurants by automating the search and booking process.

The object of the research is the process of forming and submitting proposals in restaurants.

The subject is methods and models for providing recommendations in food restaurants.

Keywords: Recommender system, chat bot, instant messaging platform, Natural Language Processing, content-based filtering, data pre-processing

ЗМІСТ

ВСТУП	9
1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ.....	11
1.1 АКТУАЛЬНІСТЬ ПРОБЛЕМИ	11
1.2 ОГЛЯД АНАЛОГІВ СИСТЕМИ.....	12
1.3 ОГЛЯД ТЕХНОЛОГІЙ ПЛАТФОРМ ОБМІНУ МИТТЄВИМИ ПОВІДОМЛЕННЯМИ	15
1.3.1 Класифікація чат-ботів	16
1.3.2 Архітектура чат-ботів	19
1.3.3 Аналіз інструментів для створення ботів	23
1.4 ОГЛЯД МЕТОДІВ СТВОРЕННЯ РЕКОМЕНДАЦІЙНИХ СИСТЕМ.....	25
1.4.1 На основі елементів подібності контенту.....	26
1.4.2 На подібності між користувачами або предметами	28
1.4.3 Гібридні методи.....	30
2 СЦЕНАРІЇ ВИКОРИСТАННЯ СИСТЕМИ.....	33
2.1 ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ ДЛЯ КОРИСТУВАЧА	33
2.2 ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ ДЛЯ АДМІНІСТРАТОРА.....	39
3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ	42
3.1 ОБЧИСЛЕННЯ ВІДСТАНІ МІЖ ЗАКЛАДОМ ХАРЧУВАННЯ ТА	
МІСЦЕЗНАХОДЖЕННЯМ КОРИСТУВАЧА	42
3.2 МЕТОД НАДАННЯ РЕКОМЕНДАЦІЙ НА ОСНОВІ КОНТЕНТНОЇ ФІЛЬТРАЦІЇ.....	42
Приклад надання рекомендацій	43
4 АРХІТЕКТУРА ТА ПРОЕКТУВАННЯ СИСТЕМИ.....	48
4.1 ЗАСОБИ РОЗРОБЛЕННЯ	48
4.2 ОПИС ПРОЦЕСУ ДІЯЛЬНОСТІ	49
4.3 ПРОЕКТУВАННЯ БАЗИ ДАНИХ	50
4.4 ПОПЕРЕДНЯ ОБРОБКА ТЕКСТОВИХ ДАНИХ.....	56

4.5	АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	57
4.6	ДІАГРАМА ПОСЛІДОВНОСТІ ДЛЯ ПРОЦЕСУ ПОШУКУ	59
4.7	РОЗРОБЛЕННЯ ІНТЕРФЕЙСУ.....	60
	Висновок до розділу.....	71
5	ТЕСТУВАННЯ СИСТЕМИ	72
5.1	РУЧНЕ ТЕСТУВАННЯ	72
5.2	МОДУЛЬНЕ ТЕСТУВАННЯ.....	78
	Висновок до розділу.....	84
6	РОЗРОБЛЕННЯ СТАРТАП ПРОЕКТУ.....	85
6.1	ОПИС ІДЕЇ ПРОЕКТУ	85
6.2	АНАЛІЗ РИНКОВИХ МОЖЛИВОСТЕЙ ЗАПУСКУ СТАРТАП-ПРОЕКТУ	87
6.3	РОЗРОБЛЕННЯ РИНКОВОЇ СТРАТЕГІЇ ПРОЕКТУ	96
6.4	МАРКЕТИНГОВА ПРОГРАМА СТАРТАП-ПРОЕКТУ	98
	Висновок до розділу	100
	ЗАГАЛЬНИЙ ВИСНОВОК.....	ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.
	ЗАГАЛЬНИЙ ВИСНОВОК.....	ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.
	ПЕРЕЛІК ПОСИЛАНЬ	102

ПЕРЕЛІК СКОРОЧЕНЬ

AI – Artificial Intelligence

API – Application Programming Interface

CBF – Content Based Filtering

CF – Collaborative Filtering

NLG – Natural Language Generating

NLP – Natural Language Processing

БД – База даних

ВСТУП

Кожного дня люди здійснюють пошук інформації в мережі Інтернет та мають необхідність працювати з великою кількістю неупорядкованої інформації, яка охоплює різні сфери життя. Часто виникає питання підбору закладів харчування, де можна провести час. Існує проблема пошуку актуальної інформації про заклад харчування, що задовольняє потреби споживача, а саме розташування закладу та його меню і цінова політика. Процес вибору закладу харчування залежить від багатьох факторів та потреб користувача.

Рекомендаційні системи – це комплекси алгоритмів, програми та сервіси, основне завдання яких передбачити, які об'єкти (товари або послуги) будуть цікаві користувачу на основі інформації про його профіль та вподобання [15]. Популярним застосуванням таких систем є інтернет-магазини різної тематики, що будуються на базі веб-сайтів або мобільних додатків. Персоналізована інформація при пошуку товару або послуги є більш ефективною у порівнянні зі звичайними пропозиціями.

Платформи обміну миттєвими повідомленнями, такі як Messenger, Telegram, Whatsapp і WeChat є популярним засобом спілкування. Люди застосовують дані платформи не тільки для обміну інформацією, а й для пошуку послуг та сервісів. Чат-боти є потужним інструментом надання інформації, який швидко набирає популярність. Процес підбору закладів харчування через чат-бот спрощується за рахунок видачі персоналізованих пропозицій та автоматизації процесу бронювання. Зазвичай процес бронювання передбачає розмову з реальною людиною та займає багато часу. При цьому середній час очікування при взаємодії з чат-ботом займає 50 секунд [1]. Інструменти для чат-ботів дозволяють автоматизувати робочі процеси та заощадити людські ресурси. Клієнти магазинів без труднощів використовують дану технологію для здійснення покупок. Прикладом є чат-бот популярної компанії Levi's, що виробляє джинсовий одяг. Чат-бот компанії допомагає користувачу підібрати одяг, визначити розмір та задати відповідні питання щодо обраного товару.

Проблему пошуку закладів харчування можна вирішити за допомогою поєднання алгоритмів рекомендаційних систем та технологій платформ обміну миттєвими повідомленнями.

Метою даної роботи є створення рекомендаційної системи пошуку закладів на основі платформи миттєвих повідомлень за рахунок автоматизації процесу пошуку закладів харчування та бронювання.

Для досягнення мети потрібно виконати наступні завдання:

- проаналізувати системи-аналоги пошуку закладів харчування;
- вибрати технології для побудови чат-ботів у платформах обміну миттєвими повідомленнями;
- визначити методи надання рекомендацій;
- розробити програмну реалізацію продукту.

1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

1.1 Актуальність проблеми

Ресторанний бізнес та напрямки, що з ним пов'язані набуває широкого розповсюдження. У сучасному світі підприємства використовують всі доступні інструменти, щоб мати зв'язок з користувачами та утримувати їх.

Розглянемо проблему пошуку потрібного закладу харчування звичайним користувачами. Сьогодні у час технологій основним інструментом пошуку потрібного місця є мережа Інтернет. Перед тим як знайти потрібний заклад користувач шукає місце за допомогою пошукових систем та картографічних сервісів. Люди стикаються з неймовірною кількістю інформації, на пошук потрібної витрачається велика кількість часу. Існує безліч видів закладів харчування, які можна порівнювати між собою за різноманітними ознаками:

- вид закладу (кафе, ресторан, бар);
- цінова категорія;
- місцезнаходження;
- направленість;
- час роботи;
- спосіб обслуговування;
- асортимент страв;
- відгуки.

Вибір закладу клієнтом формується на основі перелічених ознак. Отже, й важливо брати до уваги всі фактори вибору користувачу. При пошуку конкретної страви у пошукових системах важко знайти саме заклад харчування по даному запиту. Сервіси, що націлені на задоволення даної потреби зазвичай створені в рекламних цілях.

У рамках даної роботи вирішується проблема пошуку закладів харчування через реалізацію програмного продукту, що надає рекомендації користувачам базуючись на їх вподобаннях у швидкий спосіб.

1.2 Огляд аналогів системи

Було знайдено декілька аналогів існуючого продукту. Розглянемо функції даних аналогів.

Sirved – американський проект, що збирає відгуки з численних платформ та забезпечує користувачам на вибір неймовірну кількість нових закладів. Sirved дозволяє легко знайти свою їжу за місцем розташування, кухнею або рестораном. Зараз у додатку представлено [2] понад 400 000 ресторанів по всій Північній Америці, і він доступний безкоштовно. В основі додатка потужна пошукова система на основі меню. Продукт має такі функції:

- перегляд знайдених результатів за допомогою карти або знайдених фотографій;
- власний профіль з вподобаннями для пошуку рекомендацій;
- можливість залишати відгуки;
- створення списку місць, що подобаються користувачу;
- рейтинги з різних сайтів;
- пошук ресторанів за категоріями.

Однак клієнти не можуть зробити бронювання через додаток після пошуку ресторану на вибір. Платформа використовує машинне навчання та AI, щоб полегшити знаходження ресторанів, так що користувачі можуть здійснювати пошук у будь-який комфортний спосіб у додатку. Платформа існує, як мобільний додаток та як веб-сайт.

Google Maps and Google Places API – це відомий веб-сервіс, який надає детальну інформацію про географічні об'єкти та сайти по всьому світу. Крім звичайних карт, карти Google пропонують повітряні та супутникові види багатьох об'єктів. У деяких містах Google Maps пропонує вигляд вулиць, що містить фотографії, зроблені з транспортних засобів [3]. Також за допомогою даного сервісу можливо знайти місцеві заклади, що знаходяться поруч. Даний сервіс дозволяє переглядати відгуки відвідувачів даного закладу, фотографії місця. Повнота інформації про місце залежить тільки від власників закладів, вони додають її через свій обліковий запис.

Дані доступні за допомогою Google Places API – сервіс, який надає дані про заклади харчування, а саме [3]:

- список закладів на основі місцезнаходження користувача або рядка пошуку;
- інформація про місцевість повертає більш детальну інформацію про певне місце, включаючи відгуки користувачів;
- фотографії надає доступ до мільйонів фотографій, пов'язаних із місцем, що зберігаються в базі даних Google Place;
- заповнення місця автоматично заповнює ім'я та / або адресу місця у вигляді користувачів;
- надає послугу прогнозування запитів для текстових географічних пошуків, повертаючи запропоновані запити як тип користувачів.

Отже, даний сервіс є потужною системою, що слугує сховищем інформації про місцезнаходження закладів харчування.

HungryGoWhere. Це сайт для пошуку закладів для міста Сінгапуру. Це відомий ресурс для того, щоб дізнатися інформацію про останні відгуки та рекомендації щодо їжі та напоїв. Незалежно від того, чи потрібно вам шукати ресторан, забронювати стіл, замовити винос їжі, доставити їжу а просто прочитати про їжу, Даний сайт призначений для того, щоб допомогти користувачу заклад згідно з його бажаннями. Сайт HungryGoWhere запустився в 2006 році, як платформа для користувачів, щоб збирати інформацію про заклади та портал, на якому заклади можуть розвивати свій профіль [4].

Сервіс існує у вигляді веб-сайту, який має зручний інтерфейс. Пошук здійснюється за такими критеріями:

- по місцезнаходженню вказавши вулицю міста;
- по конкретному закладу харчування;
- по конкретній назві страви.

OpenTable – це сервіс, що займається бронюванням ресторанів. У 1999 році [4] веб-сайт почав працювати, пропонуючи бронювання обмеженого вибору ресторанів

у Сан-Франциско. Зараз система має бота в платформі обміну миттєвими повідомленнями Facebook Messenger.

Ресторани, які підписалися на послуги сервісу, використовували програмне забезпечення компанії для обробки бронювання, зроблених на веб-сайті, в результаті чого система бронювання працює в реальному часі, як для закусочних, так і для ресторанів. Послуга з цього часу розширилася на 30 000 ресторанів у більшості штатів США, а також у кількох великих міжнародних містах. Бронювання можна зробити онлайн через веб-сайт та бот [5].

Сайт (рисунок 1.1) надає можливість пошуку закладів за допомогою місцезнаходження та на основі інформації про вільні місця в закладі. Також є можливість переглянути дані про заклади та меню.

Платформа має потужний API на Java, що надає доступ до інформації про заклади включаючи ціни, місцезнаходження, відгуки та посилання на систему бронювання.

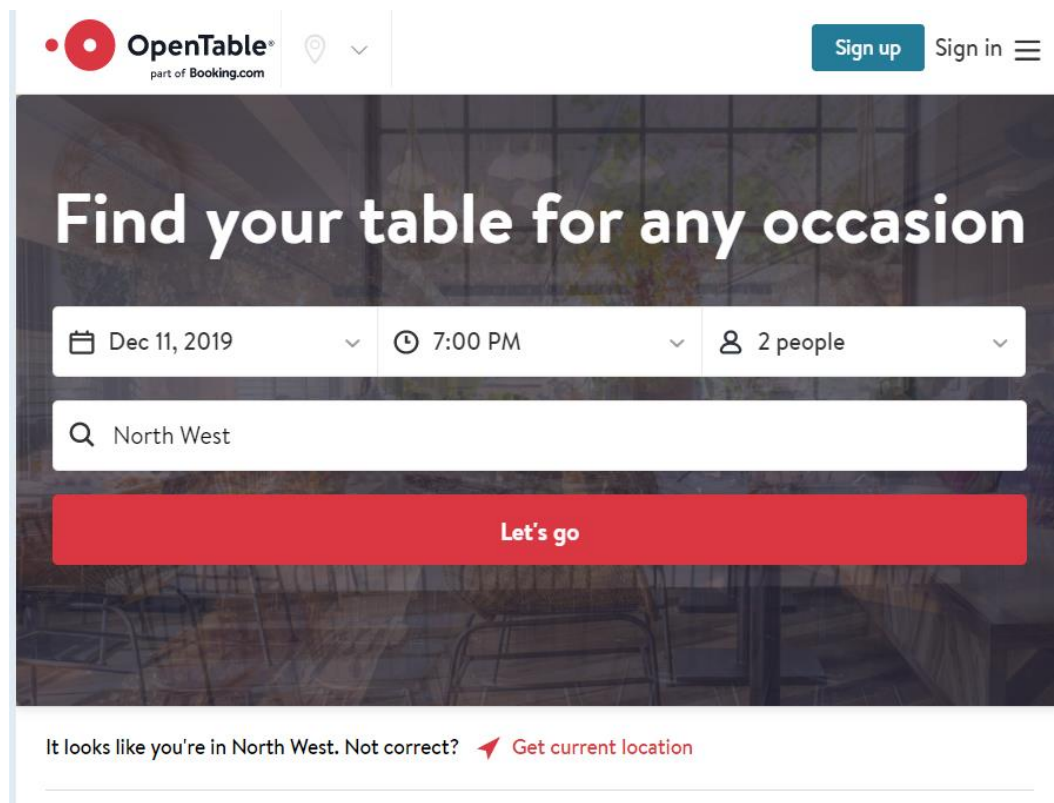


Рисунок 1.1 – Інтерфейс для пошуку OpenTable

При пошуку аналогів виявилось, що програмних продуктів у сфері ресторанного бізнесу, які пов'язані з автоматизацією процесів бронювання столику, пошуку страв та замовлення доставки існує у великій кількості.

Порівнюючи сервіси можна зробити висновок, що аналоги переважно використовують веб-сайти для реалізації їх функціоналу та не відповідають вимогам системи, що розроблюється. Основною відмінністю є те, що продукт розроблений на основі платформи обміну миттєвими повідомленнями та частково не співпадає за функціоналом та способом реалізації жодному з розглянутих аналогів.

1.3 Огляд технологій платформ обміну миттєвими повідомленнями

Миттєві повідомлення (Instant Messaging), найпопулярніше втілення технології синхронного обчислення для текстового чату серед користувачів. Опитування показало, що третина користувачів комп'ютерів використовує чат на роботі, щоб підтримувати зв'язок із колегами та клієнтами. Крім того, підраховали, що буде 250 мільйонів акаунтів чатів, включаючи бізнес-рахунки через 10 років[1]. Завдяки своїй функціональності для взаємодії, ІМ підтримує неформальну комунікацію на робочому місці, де електронна пошта, телефон і факс вже широко прийняті, але також полегшує деякі процеси, які роблять можливим ухилення від загального поширення непотрібної [5] інформації.

Сьогодні існують такі популярні платформи обміну миттєвими повідомленнями, як Telegram, Viber, WhatsApp. Вони слугують потужним інструментом для спілкування та є найпоширенішими між користувачами у наш час. Розробники намагаються зробити дані платформи настільки уніфікованими, щоб кожен користувач мав можливість здійснювати велику частину роботи кожного дня саме в одному додатку, а не перемикаючись між різними додатками. Наприклад, здійснювати облік витрат, відповідати на запити щодо підтримки клієнтів або навпаки, перевіряти статус замовлення товарів у магазині. Одним із зручних способів реалізації даних можливостей є чат-боти.

1.3.1 Класифікація чат-ботів

Чат-боти – програма, яка імітує розмову з людьми-користувачами, використовуючи текст, голос або зображення або поєднання розмовної та візуальної евристики [7]. Це спілкування зі змодельованою комп'ютерною програмою. Саме дана технологія заощаджує час та зусилля, автоматизуючи підтримку клієнтів. Прогнозують, що в наступні 10 років понад 85% взаємодій з клієнтами здійснюватимуться без участі людини [1]. Розповсюдження чат-ботів пов'язано зі змінами в комунікаціях між компаніями та користувачами. Компанії адаптують канали зворотного зв'язку за допомогою веб-сайтів та додатків переважно на мобільних пристроях.

Сьогодні ринок мобільних додатків є великим, та конкуренція в даному напрямку зростає. Користувачі надають перевагу простим та інтуїтивно зрозумілим інтерфейсам тому й зростає популярність чатів, що існують як додатки обміну миттєвими повідомленнями. Зазвичай чат-боти використовують, як приємний та легкий спосіб побудови діалогу для користувачів.

Якщо порівнювати звичні для людей способи комунікацій, а саме телефон або електронну пошту, чат-боти – доступніше та надають відповідь користувачам миттєво. Процес розроблення чат-ботів є достатньо швидким. Існує безліч інструментів, що полегшують розробку ботів.

Безліч популярних сервісів створюють чат-ботів для підтримки комунікації зі своїми клієнтами. Створення чат-бота дешевше та швидше у реалізації, ніж розробка повноцінної платформи. З появою даної технології можна зберегти людські ресурси та позбавитись від такої проблеми як неуважність при роботі. Чат-бот надає цілодобову підтримку, завжди готовий до швидкої взаємодії з клієнтами та вирішує поширені недоліки, що є в сайтах(важка навігація по веб-сторінках, складність пошуку відповідей на легкі питання, витрати часу на пошуку сервісів).

Сьогодні існує безліч видів чат-ботів, що відрізняються один від одного технологіями реалізації, тематикою, процесом взаємодії, тощо. Наведемо класифікацію чат-ботів за такими ознаками, як направленість, ціль та метод генерації відповідей [6] (рис. 1.2).

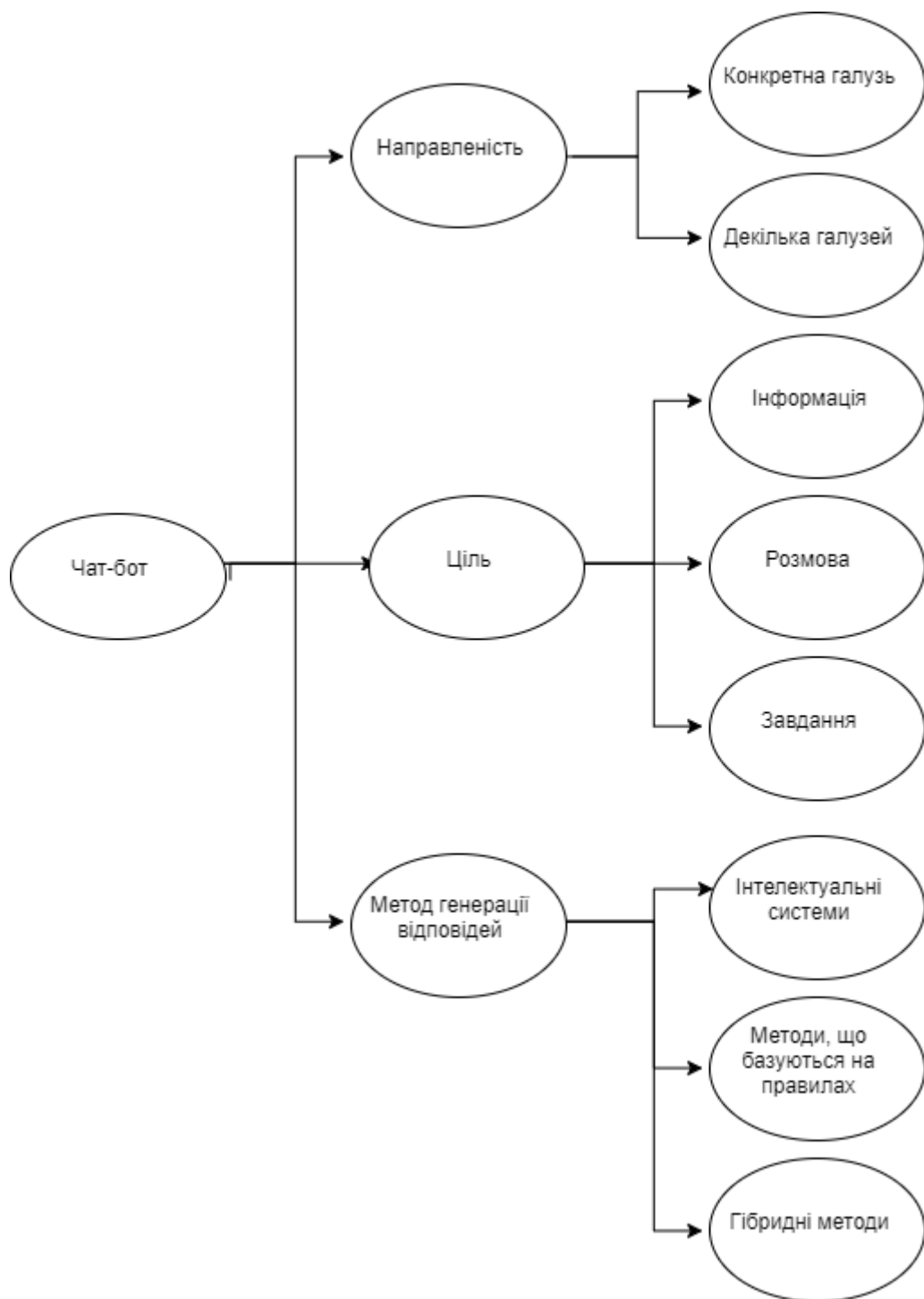


Рисунок 1.2 –Класифікація чат-ботів

Ціль є найважливішою з ознак для створення чат-ботів. Розглянемо більш детально кожну з них.

Боти призначені для надання користувачеві інформації, яка зберігається заздалегідь або доступна з конкретного джерела. Зазвичай вони базуються на алгоритмах пошуку інформації і отримують результат запиту зі сховища даних, або

перевіряють відповідність рядків. Здебільшого вони посиляються на статичне джерело інформації, наприклад, сторінку поширених запитань на сайті або на базу даних фільмів. Прикладами таких чат-ботів є чат-боти для відповідей на поширені запитання.

Чат-боти, що ведуть розмову з користувачем. Їх мета правильно реагувати на запити користувача. Отже, часто побудовані з метою продовження розмови з користувачем на основі таких методів, як перехресне опитування, ухилення від перешкоджання [6].

Чат-боти, що виконують завдання. На основі завдань вони виконують певну дію, наприклад, забронювати столик в ресторані або рекомендують товари. Прикладом є боти для пошуку нерухомості.

Також боти можна класифікувати за методом обробки вводу, а саме яким чином чат-бот буде реагувати на запити користувача [1, 8]. Інтелектуальні системи генерують відповіді і використовують розуміння природної мови для розпізнавання запиту. Ці системи використовуються, коли направленість бота є вузькою, і для розробки системи існує достатньо даних.

Системи на основі правил використовують відповідність шаблонів і мають обмеження. Вони можуть бути використані, коли кількість можливих результатів чітко визначена і всі сценарії можна передбачити.

Гібридні системи є сукупністю систем, що базуються на правилах та алгоритмах машинного навчання [12]. Прикладом може бути система, яка використовує блок-схему для управління розмовою. Відповіді, які генеруються за допомогою здійснюються за допомогою обробки природних мов (natural language processing). При цьому боти повинні належати виключно до однієї тематики.

Належність до конкретного типу існує у кожного бота в різних пропорціях. Наприклад, всім ботам потрібні певні інструменти спілкування в чаті, боту для інтернет-магазину потрібно буде використовувати обробку запиту інформації при потребі пошуку відповіді до поширених запитань, або шукати за допомогою бази даних, коли потрібно знайти конкретний продукт. Метод розроблення бот, може включати поєднання декількох алгоритмів. Також направленість, або його тематика

може поширюватися на одну галузь так і бот може поширюватися на декілька галузей для надання сервісу. Наведемо ключові фактори ефективної роботи чат-ботів [11].

Спілкування. Здатність чат-бота вдало вести розмову з користувачем. Це може бути голосовий або текстовий режим. Також, важливо персоналізувати розмову через аналіз спілкування та настрою користувача.

Розуміння. Програмі необхідно осмислювати людські запити, а саме вміти вилучати контекст, настрій, розуміння напрямку у розмові.

Співпраця. Бот повинен постійно вчитися коректно оброблювати запити користувача на основі досвіду інших джерел даних та пристроїв.

1.3.2 Архітектура чат-ботів

Розглянемо детальніше елементи, що входять до архітектури чат-бота. Нижче на рисунку 1.3 наведена загальна схема що містить структуру компонентів для побудови чат-ботів [7].

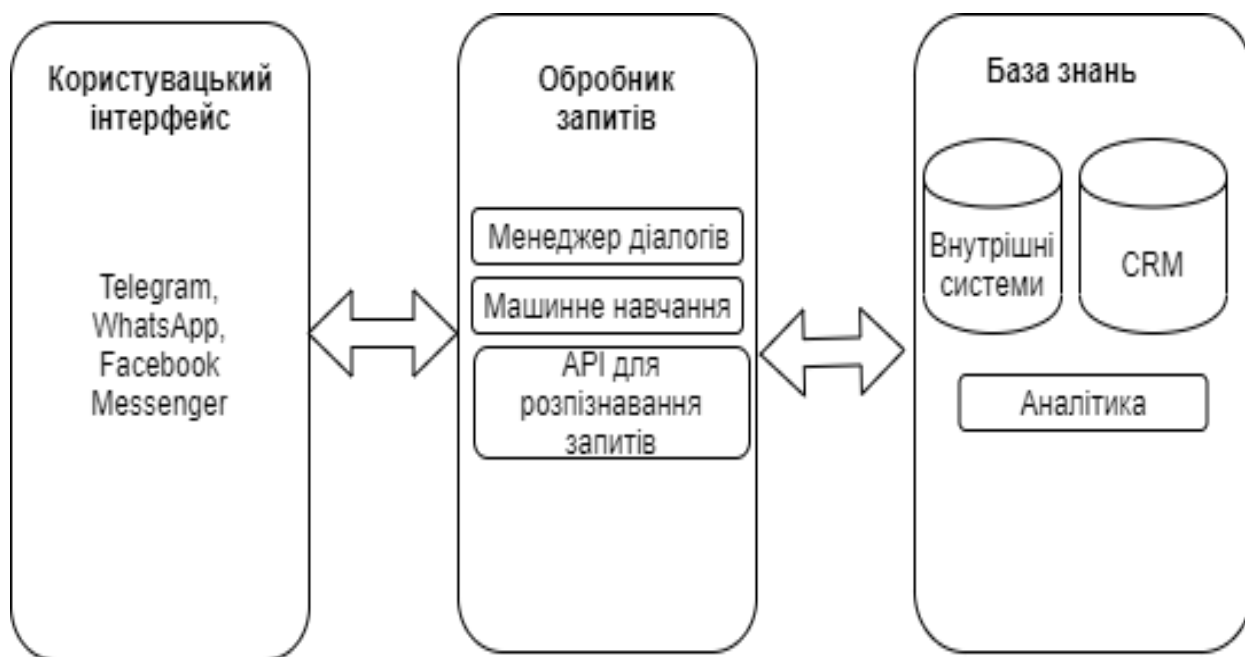


Рисунок 1.3 – Архітектура чат-ботів

Розглянемо кожен з компонентів окремо.

Користувацький інтерфейс. Для того, щоб скористуватися чат-ботом користувачу обов'язково потрібно бути авторизованим у платформі обміну

миттєвими повідомленнями в якій створений чат-бот. Користувацьким інтерфейсом зазвичай слугує додаток, або комп'ютерна версія додатку. Прикладами таким додатків є Telegram, WhatsApp, Facebook Messenger, тощо. Вхідною інформацією від користувача можуть слугувати: текстові та голосові запити, інформація з кнопок.

Обробник запитів. Кожен запит, який надсилається до боту переходить в обробник запитів, що є програмою, яка розпізнає текстові запити від користувача різними способами.

Для того, щоб методи пошуку були точними потрібно виключити можливість помилок за допомогою попередньої обробки тексту. Зазвичай вона включає в себе такі кроки:

- перевірка орфографії;
- поділ на речення;
- маленькі/великі літери;
- вилучення літер та символів, що заважають процесу пошуку;
- розпізнавання шаблонів;
- лемматизація слів;
- пошук синонімів.

Після попередньої обробки тексту результатом буде список слів або речень, які потрібно класифікувати. Для кожного виду запитань повинен існувати унікальний шаблон у базі даних

Обробка природньої мови або Natural Language Processing [11] допомагає розпізнати значення зі складних мовних конструкцій. Наприклад, коли користувач надсилає чат-боту складне запитання. Дана технологія спрощує процес обробки та значно покращує відповіді чат-боту на складні запитання. Обробка природньої мови має три поняття: сутність, контекст та наміри. Сутність – представляє суб'єкт в чатах. Наприклад, платіжна система для оплати послуг в чаті. намірами слугує взаємодія користувачів в чаті. Наміри – це дії користувача, що повинні розпізнаватися чат-ботом. Потреба користувача може виражатися різними текстовими запитаннями, але мати однакові наміри. Тобто, питання «Чи є у вас меню з конкретного закладу харчування?» та «Чи є можливість забронювати столик в даному закладі?» будуть

розглянуті як однакові наміри. У свою чергу контекст легко співвідносить наміри користувача між собою. Наприклад, якщо бронювати столик в закладі харчування, тобто надіслати боту запит на бронювання, а потім ввести дату бронювання, чат-бот повинен співвідносити подані дані, а саме те, що дата пов'язана з етапом бронювання.

NLP надає кращі результати для відповідей та складних запитань за рахунок компоненту навчання. Цей процес може містити такі етапи:

- аналіз настроїв допомагає боту визначити, чи користувач має досвід роботи з чат-ботом;
- поділ речень на слова та присвоєння їм категорій, наприклад, назва продукту чи адреса;
- нормалізація речень через виправлення помилок;
- пошук залежностей між словами у тексті.

Після всіх зазначених етапів можливе використання Natural Language Generation, що створює відповідь для користувача залежно від її контексту. Види відповідей для користувача поділяються на[10]:

- статичні відповіді, що вибираються з визначеного діапазону відповідей (наприклад, «за даним запитом знайдено 20 результатів»);
- динамічні відповіді використовують базу знань, щоб проаналізувати правильну відповідь;
- відповіді, що згенеровані базуються на алгоритмах глибокого навчання для покращення відповідей.

База знань. Також існують додаткові сервіси, що розширюють функціонал чат-бота та забезпечують його повноцінну роботу. Зазвичай, це звичайні бази даних, що зберігають дані, які необхідні для повноцінної роботи. Текстові дані для користувача можуть аналізуватися та бути використані для покращення розуміння потреб користувачів. Метрики для аналізу наводяться в таблиці 1.1.

Таблиця 1.1 – Метрики для аналізу використання чат-боту

Метрика	Значення	Частина взаємодії з продуктом
Коефіцієнт утримання	Кількість користувачів, які повертаються до використання чату-боту заданий часовий проміжок.	Чат-бот
Коефіцієнт виконання цілі	Кількість успішної роботи ботів, скільки запитів було виконано успішно, чи надавали правильну інформацію.	Чат-бот
Кількість користувачів	Визначає загальну кількість користувачів.	Користувач
Заохочення користувачів	Користувачі оцінюють досвід використання продукту.	Чат-бот
Пропущені повідомлення	Кількість повідомлень, що дають незадовільну відповідь користувачу	Аналіз повідомлень
Активні користувачі	Кількість користувачів, що повертаються до використання чат-боту	Користувач

Чат-бот може використовуватись, як система взаємодії з клієнтами (Customer Relationship Management) у вигляді інтерфейсу. Наприклад, приймати замовлення на бронювання столика від користувачів можна при наявності чат-бота, який повідомляє про активні бронювання столиків від системи, що приймає замовлення на бронювання.

Отже, чат-бот є потужним інструментом для комунікації з користувачами та має багато переваг перед традиційними технологіями. Для того, щоб розробити чат-

бот потрібно обрати платформу для обміну повідомленнями, яка має відповідну аудиторію, визначити тематику та механізм роботи.

1.3.3 Аналіз інструментів для створення ботів

Інтерфейсом для ботів може слугувати переважно більшість платформ обміну миттєвими повідомленнями. Розглянемо інструменти для створення чат-ботів серед таких додатків як Facebook, Telegram, Viber, WhatsApp які популярні в Україні.

Messenger. Платформа надає безкоштовний інтерфейс для створення ботів. Інтеграція з даним сервісом автоматично зберігає дані про користувачів та поточні чати. Бот автоматично зберігає їх і дозволяє повторно використовувати їх у своїх сценаріях, а саме таку інформацію як аватар, мову користувача, часову зону. Ці дані дозволяють підлаштовуватися під мову користувача та час під час розробки. Також існують такий формат відображення даних Carousel, який дозволяє відображати декілька зображень та відео в одному повідомленні [11]. Недоліком реалізації даного інструмента є те, що для того, щоб отримати дозвіл на його створення потрібно узгоджувати з адміністраторами сайту та підтверджувати актуальність та коректність інформації, що розповсюджує бот.

Спілкування з людьми через чат-бот Messenger відрізняється від взаємодії через сторінку Facebook. База даних не підключається до боту та отримується доступ через додаток для створення ботів, наприклад, Chatfuel або ManyChat. Можливість спілкуватися можлива тільки з знайомими контактами зі сторінки соціальної мережі.

Viber. Платформа відкрита і безкоштовна для розробників для створення чат-ботів у системі, вона надає зручний інтерфейс. Viber використовує API бот Viber Chat і створює приватний зашифрований канал зв'язку між ботом і підписаним користувачем Viber [9]. Всю інформацію, передану в цьому каналі, можуть бачити лише підписаний користувач Viber та адміністратори бота. Для створення бота потрібен активний акаунт та підтвердження тематики бота зі сторони адміністраторів платформи.

Боти можуть відправляти повідомлення, медіа файли, координати місцезнаходження, відео файли, контакти, стікери, клавіатуру. Розробник бота має доступ до такої інформації користувача:

- ім'я;
- країна;
- мова;
- тип та марку пристрою;
- його статус.

Боти доступні на таких пристроях, як IOS та Android та на комп'ютерній версії додатку.

Telegram. Боти є сторонніми програмами, які працюють в додатку. Користувачі можуть взаємодіяти надсилаючи їм команди, повідомлення та запити. Функції, які можуть використовуватися в чат-боті:

- можливість інтеграції з платіжними системами;
- реалізація ігор;
- інтеграція з іншими сервісами. Прикладами є Gmail Bot, IMDB Bot, GitHub Bot;
- створення власних інструментів, а саме надання сповіщень, перекладів, форматування.

На відміну від інших платформ, особливістю Telegram є те, що він використовує хмарні сховища з безперебійною синхронізацією [10]. Це надає користувачу такі переваги, як доступ до повідомлень з декількох пристроїв одночасно, включаючи планшети та комп'ютери, та можливість поширювати необмежену кількість фотографій, відео та файлів. Всі дані доступні в хмарі у будь-який час та не обов'язково мають зберігатися на телефоні користувача. Завдяки інфраструктурі та шифруванню даних, Telegram є більш швидким та безпечним. Крім того, Telegram є у безкоштовному доступі і залишатиметься безкоштовним без використання реклами.

Створення чат-ботів у платформі здійснюється у легкий спосіб, якщо мати досвід програмування на будь-якій мові. Інформацію, яку використовує бот про користувача не відрізняється від інформації, що бачить кожний користувач. Telegram.

Чат-бот не має доступу до статусу користувача та до його номеру телефону. Вони працюють у двох режимах: як окремий приватний чат для кожного користувача та, як бот у спільних групових чатах.

1.4 Огляд методів створення рекомендаційних систем

Існує великий клас програм, які передбачають запити користувачів. Рекомендаційні системи є такими програмами, які намагаються підібрати конкретним користувачам найбільш необхідні елементи (продукти чи послуги) шляхом прогнозування інтересів користувача до предмета на основі пов'язаної інформації про товари, про вподобання користувачів та взаємодії між предметами та користувачами [14].

Метою розробки таких систем є зменшення перевантаження інформацією шляхом вилучення найменш актуальної інформації з величезної кількості даних, забезпечуючи тим самим персональні послуги. Види рекомендаційних систем зображені на рисунку 1.4.



Рисунок 1.4 – Види рекомендаційних систем

Найважливішою особливістю системи рекомендацій є можливість передбачити переваги та інтереси користувача, аналізуючи поведінку цього користувача та/або поведінку інших користувачів, створивши персоналізовані рекомендації. Наведемо приклади таких систем [15]:

- підбір статей новин для читачів електронних журналів та газет на основі прогнозу читацьких інтересів;
- підбір клієнтам роздрібних пропозиції щодо того, що вони хотіли б придбати, виходячи з історії покупок або схожих товарів.

Найпоширенішими сценаріями використання даних методів є прогнозування, ранжування та класифікація, які проявляються в різних типах інтерфейсів користувача.

У завданні прогнозування основна увага приділяється передбаченню рейтингів для необмежених предметів користувача і показуючи йому ці прогнози. Наприклад, сайт з фільмами зазвичай надає користувачу прогнозовані рейтинги фільмами, щоб допомогти йому визначитися з вибором. Дані системи використовують ряд різноманітних технологій. Класифікуємо методи реалізації систем на три групи.

1.4.1 На основі елементів подібності контенту

Наприклад, якщо користувач переглянув багато фільмів про інопланетян, система порекомендує йому фільм, віднесений до бази даних як такий, що має жанр «фантастика».

У системі, що базується на вмісті, для кожного елемента створюється профіль, який є записом або колекцією записів, що представляють важливі характеристики цього елемента [21]. У простих випадках профіль складається з деяких характеристик предмета. Розглянемо особливості закладів харчування, які можна розглянути для створення рекомендацій.

1. Місцезнаходження. Відвідувачі найбільшу надають перевагу найменшій віддаленості закладу від поточного місцезнаходження. Якщо заклад харчування буде в іншому кінці міста, то відвідувач не надасть йому перевагу.

2. Цінова політика. Відвідувачам важливо дізнатися про цінову політику закладу, щоб оцінити та розрахувати свої кошти.
3. Тип кухні. Наприклад, деякі відвідувачі віддають перевагу стравам лише української кухні, інші – корейської або грузинської. Чи хтось не вживає м'ясо та шукає тільки вегетаріанський заклад.
4. Часи роботи. Деякі відвідувачі встають рано і їм важливо обрати заклад, що відкривається о восьмій.
5. Спосіб оплати. Деякі розраховуються тільки в безготівковій формі, а інші мають тільки готівку.

Існує багато й інших властивостей для закладів харчування, які також можуть бути використані. Матриця подібності для даного методу зображена на рисунку 1.5.

Даний алгоритм не потребує даних про користувача. Дана перевага дає можливість до легкого масштабування кількості користувачів. Оскільки інформація про смаки інших користувачів відсутня, існує можливість легко робити рекомендації людям з особливими вподобаннями.

Недоліком даного методу є те, що даний метод вимагає від розробника знати багато інформації про специфіку товарів, що пропонується.

		Елементи				
		Продукт 1	Продукт 2	Продукт 3	Продукт 4	Продукт 5
Елементи	Продукт 1	1	0.8	0.578	0.5	0.1
	Продукт 2	0.8	1	0.04	0.678	0.03
	Продукт 3	0.578	0.04	1	0.005	0.2
	Продукт 4	0.5	0.678	0.005	1	0.4
	Продукт 5	0.1	0.03	0.2	0.4	1

Рисунок 1.5 – Матриця подібності

1.4.2 На подібності між користувачами або предметами

Колаборативна фільтрація – метод надання рекомендацій в основу своїх методів досвід користувачів, шукаючи користувачів з подібними інтересами, а потім на основі цих даних підбирає рекомендації. Даний метод включає три підходи: для того, щоб метод працював користувачі повинні брати участь у побудові рекомендаційної системи;

- надати користувачам простий доступ до оцінки продуктів;
- в результаті рекомендації повинні відповідати дійсним вподобанням користувачів.

Основна ціль даного методу побудови полягає в тому, що рекомендації персоналізуються. В обчислювальному вигляді формат даних є простим та рівномірним. Існує поняття матриці елементів користувача, комірки якої представляють рейтинги від кожного користувача для кожного фільму. Якщо оцінки від користувача не існує, то значення залишається пустим. Алгоритм дій користувача є досить простим.

Користувач висловлює свої вподобання за рейтинговими позиціями (наприклад, журнали, навушники або іграшки), які надає система. Потім рейтинги є основою, що формують вподобання в даній галузі.

Система порівнює ці рейтинги з іншими рейтингами, поданими всіма іншими користувачами системи, результатом є набір однодумців (людей, що мають схожі смаки). Система рекомендує предмети, які оцінили найближчі сусіди, а саме ті, що не були оцінені. Ключовим питанням залишається те, як поєднати та зважити вибір ваших однодумців. З часом система набуває все більш точного представлення ваших вподобань.

Колаборативна фільтрація [15] має багато застосувань у вирішенні різноманітних задач. На сайтах, таких як Amazon є рекомендаційні центри, де пропонуються огляди експертів для товару. Також користувачі можуть оцінювати товари самостійно, а потім отримувати персоналізовані рекомендації, розраховані даним механізмом фільтрації. Вподобання користувача, а саме алгоритм його дій на

сайті. Наприклад, придбання навушників може сприйматися, як доказ зацікавленості не лише у самому продукті, а й у марці, що продає даний товар.

Колаборативна фільтрація не просто підтримує існуючу діяльність, вона вимагає від користувачів брати участь у новій обчислювальній діяльності. Ця діяльність має єдину комбіновану роль, яка шукає рекомендації та надає перевагу. Усі виконують ту саму роботу і отримують персоналізовані результати (отримують рейтингові позиції, як рекомендації). Рівномірність ролей має як хороші, так і погані аспекти. З одного боку, спостережувана практика говорить про те, що більшість людей не хочуть пропонувати рекомендації; натомість вони просто хочуть ними скористатися. З іншого боку, рейтингові позиції не є особливо тяжкою роботою, і користувач виконує цю роботу саме тоді, коли хоче отримати рекомендацію.

Зазвичай між постачальником рекомендацій та одержувачем не повинно бути контактів. Але проектувальники систем можуть використати цей метод для рекомендацій людей зі схожими вподобань. Це є методикою формування соціальної мережі, оскільки людей можна автоматично зв'язувати з іншими зі схожими вподобаннями.

Існує декілька технічних проблем [16] та недоліків для алгоритмів колаборативної фільтрації.

1. Для того, щоб сформувавши рекомендації в системі повинно бути достатньо користувачів. Жодна рекомендація щодо товару не може бути запропонована, поки їй не наддадуть оцінку.
2. Більшість користувачів не оцінюють більшість елементів, отже, матриця елементів-користувачів зазвичай дуже розріджена. Якщо рекомендовано багато предметів, навіть, якщо вони наявні, багато користувачів, оскільки матриця оцінок користувачів є розрідженою, важко знайти користувачів, які оцінювали ті самі предмети.
3. Не можна рекомендувати предмет, якого раніше не оцінювали. Ця проблема в основному стосується нових продуктів. Продукт з нечіткими характеристиками також може зіткнутися з даною проблемою.

CF не може рекомендувати предмети тому, хто має унікальні нетипові смаки. У цьому випадку система буде надавати користувачу популярні товари або послуги. Матриця подібності для даного методу зображена на рисунку 1.6.

		Елементи				
		Продукт 1	Продукт 2	Продукт 3	Продукт 4	Продукт 5
Користувачі			5		1	
			1			
		4				
				3	4	3

Рисунок 1.6 – Матриця подібності для колаборативної фільтрації

Отже, для того, щоб використати даний метод в системі повина бути передбачена функція для оцінювання товарів кожним користувачем.

1.4.3 Гібридні методи

Існують різні спроби поєднання методів, що базуються на контенті та колаборативній фільтрації залежно від наявності даних та вимог. Гібридна система рекомендацій – це поєднання системи, що включає різні алгоритми рекомендаційних систем.

Деякі гібридні системи базуються на поєднанні алгоритмів. Наприклад, поєднання результатів із контентного метода та метода колаборативної фільтрації. Системи, що мають монолітну конструкцію створюються на основі інтегрованих алгоритмів рекомендацій з різними типами даних. Між ними в різних частинах

алгоритму не може існувати чітких правил, такий алгоритм прагне до вільного поєднання різних джерел даних. Гібридні системи можна поділити на види [18].

1. Зважені системи. Всі метрики оцінки системи об'єднуються у єдину зважену оцінку. Можуть використовувати як і евристичні, так і формальні статистичні моделі.

2. З розширенням функцій. Результати підбору однієї системи рекомендацій використовується для створення вхідних даних для інших систем. На відміну від каскадних систем, що послідовно уточнюють рекомендації з попередніх систем, підхід з розширенням функцій розглядає функції як вхідні дані для запуску наступної системи. Результати одного класифікатора використовуються як функції для наступного.

3. Системи, що змінюють алгоритм. Алгоритм системи перемикається між різними рекомендаційними системами залежно від потреб. Наприклад, використання системи, що базується на знаннях на початкових етапах для уникнення проблеми холодного старту, а на пізніших етапах переключення на інший алгоритм, що працює точніше.

4. Каскадні. У даному випадку одна рекомендаційна система уточнює рекомендації у іншій системі. Навчання рекомендаційної системи проводиться на базі інших систем, а потім результати узагальнюються.

5. З комбінаціями функцій. У цьому випадку функції різних джерел даних поєднуються і використовуються в контексті єдиної системи рекомендацій.

6. Змішані. Рекомендації з декількох двигунів одночасно подаються користувачеві. Підхід досить відрізняється від усіх вищезгаданих методів. Він використовує інші рекомендації як чорні скриньки, але це не поєднує прогнозовані рейтинги одного і того ж предмета від різних рекомендацій.

Зовсім інший підхід до рекомендаційної системи включає використання інформації їх профілів та матриць подібності полягає в тому, щоб розглядати проблему як задачу машинного навчання. Зазначити дані, як навчальний набір, а для кожного користувача створити класифікатор, який передбачає рейтинг всіх елементів. Дерево рішень – це сукупність вузлів у вигляді бінарного дерева. Листя

визначають результат; у такому випадку рішення буде "подобається" або "не подобається" [19]. Кожен внутрішній вузол є умовою для класифікованих об'єктів; умова буде предикатом, що включає одну або кілька особливостей елемента.

Отже, існує неймовірна кількість рекомендаційних систем, що відрізняються методами реалізації, цілю створення, даними, що аналізуються. Оскільки система для підбору закладів харчування бере за увагу саме інформацію про заклади, то найкращим алгоритмом для надання рекомендацій буде алгоритм на основі контенту. Це метод буде аналізувати заклади харчування між собою та підбирати потрібні для користувача варіанти.

Висновок до розділу

У розділі були проаналізовані існуючі рішення для побудови рекомендаційної системи, а саме їх класифікацію.

Детально була описана архітектура чат-ботів та інструменти їх створення за допомогою відомих платформ обміну миттєвими повідомленнями.

Були наведені аналоги для існуючого продукту та було розглянуто їх функціональні можливості.

2 СЦЕНАРІЇ ВИКОРИСТАННЯ СИСТЕМИ

Перш ніж розпочати роботу над програмною реалізацією системи, необхідно чітко зрозуміти перелік вимог, що повинні задовольнятися системою. У додатку наведено структурну схему варіантів використання.

Системою будуть користуватися два актори: користувач чат-бота та адміністратор. Користувач – це актор, який взаємодіє з інтерфейсом чат-бота та виконує функцію пошуку закладів та здійснює бронювання. Адміністратор взаємодіє з інтерфейсом сайту, що надає йому можливості змінювати інформацію про заклад та переглядати модуль статистики.

2.1 Діаграма варіантів використання для користувача

Розглянемо всі випадки використання системи (прецеденту) для користувача. Всі випадки для пошуку закладів зображені на рисунку 2.1. Основною функцією, що здійснює користувач є пошук закладів харчування та бронювання. Опис прецеденту «Знайти заклад» наведений у таблиці 2.1.



Рисунок 2.1 – Варіанти використання для користувача при пошуку закладів

Таблиця 2.1 – Опис прецеденту «Знайти заклад»

Назва	Знайти заклад
Актор	Користувач
Опис	Система готова до пошуку пропозицій
Передумови	Користувач знаходиться у головному меню
Результат	Користувачу буде запропоновано вибрати спосіб пошуку результатів

Перед тим як знайти заклад користувач має надіслати запит (табл. 2.2) з його вподобаннями або надіслати розташування (табл. 2.3) для подальшого пошуку.

Таблиця 2.2 – Опис прецеденту «Надіслати локацію»

Назва	Надіслати місцезнаходження
Актор	Користувач
Опис	Надіслати координати для опрацювання. місцезнаходження користувача як параметр для пошуку.
Передумови	Користувач повинен бути в головному меню або здійснити новий запит.
Результат	Користувачу буде запропоновано результати пошуку відповідно до попередніх параметрів пошуку

Таблиця 2.3 – Опис прецеденту «Надіслати запит»

Назва	Надіслати запит
Актор	Користувач
Опис	Надіслати текстовий запит для подальшого пошуку пропозицій.
Передумови	Користувач може зробити запит на будь-якому етапі, окрім бронювання.
Результат	Користувачу буде запропоновано кількість результатів для перегляду або можливість здійснити фільтрацію результатів по місцезнаходженню.

Після того, як користувач здійснив пошук він може переглянути результати пошуку. Прецедент «Переглянути результати» наведений у таблиці 2.4.

Таблиця 2.4 – Опис прецеденту «Переглянути результати»

Назва	Переглянути результати
Актор	Користувач
Опис	Перегляд результатів останнього пошуку. Результати містять інформацію про заклад.
Передумови	Користувач повинен ввести дані для пошуку.
Результат	Користувачу будуть відображені результати пошуку з можливістю збільшити число результатів.

Також він може обрати конкретний заклад да дізнатися про нього інформацію (табл. 2.5).

Таблиця 2.5 – Опис прецеденту «Перегляд інформації про заклад»

Назва	Перегляд інформації про заклад
Актор	Користувач
Опис	Перегляд детальної інформації про заклад відповідно до вибору користувача.
Передумови	Користувач здійснив пошук, переглянув пропозиції, обрав конкретний результат.
Результат	Користувачу буде представлена веб-сторінка з інформацією про заклад.

Також користувач має можливість бронювання столу обраному закладі, опис прецедента описана у таблиці 2.6.

Таблиця 2.6 – Опис прецеденту «Забронювати стіл»

Назва	Забронювати стіл
Актор	Користувач
Опис	Можливість зробити бронювання столику у закладі будь-якого результату пошуку. Для того щоб успішно забронювати стіл потрібно вказати час, дату, кількість гостей та номер телефону.
Передумови	Користувач в результатах пошуку обрав заклад, що потрібно забронювати.
Результат	Для даного користувача буде створено бронювання для подальшого опрацювання та з'явиться можливість скасування бронювання.

У разі не вдалого бронювання закладу у користувача існує можливість відмінити поточне бронювання (табл. 2.7).

Таблиця 2.7 – Опис прецеденту «Скасувати бронювання»

Назва	Скасувати бронювання
Актор	Користувач
Опис	Можливість скасувати існуюче бронювання.
Передумови	Користувач повинен мати активні бронювання.
Результат	Бронювання буде відмінено, користувач може зробити нове бронювання.

Фрагмент діаграми використання для процесу бронювання зображений на рисунку 2.2. Опишемо варіанти використання даного процесу у таблицях 2.8-2.14.

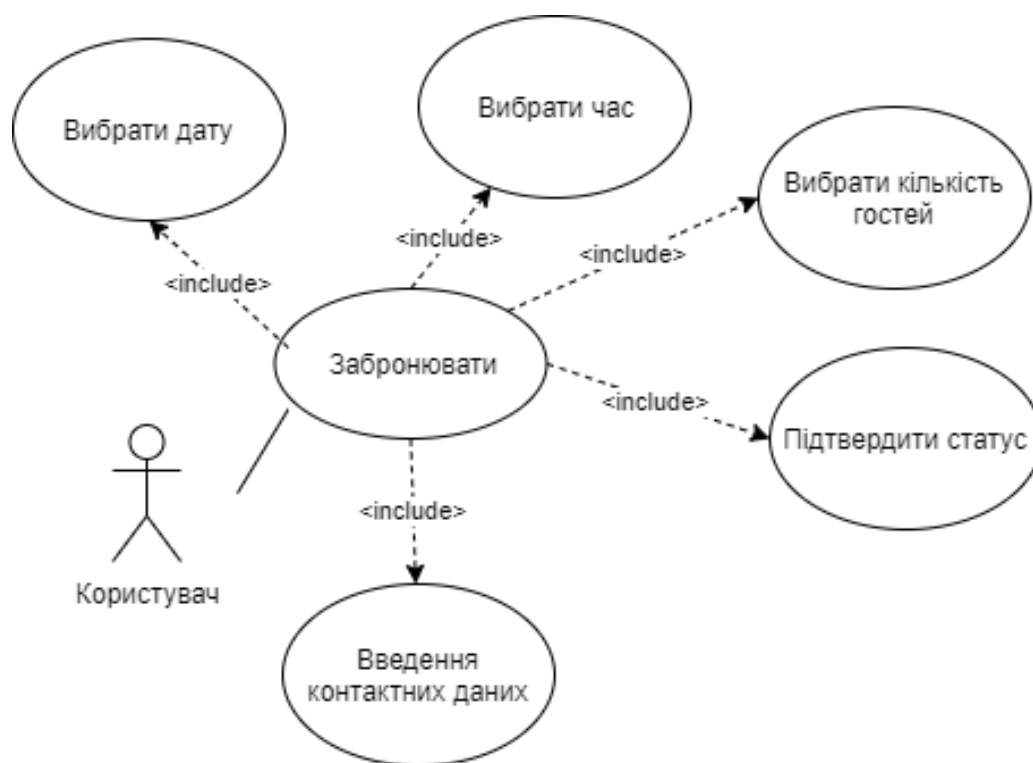


Рисунок 2.2 – Варіанти використання для користувача для бронювання

Таблиця 2.8 – Опис прецеденту «Вибрати дату»

Назва	Вибрати дату
Актор	Користувач
Опис	Обрати актуальну дату бронювання для закладу харчування.
Передумови	Обрати заклад для подальшого бронювання
Результат	Система підтвердить дату бронювання та запропонує вибрати наявний час для бронювання у вибрану дату.

Таблиця 2.9 – Опис прецеденту «Вибрати час»

Назва	Вибрати час
Актор	Користувач
Опис	Обрати актуальний час бронювання для закладу харчування.
Передумови	Обрати дату для визначення наявного часу для подальшого вибору.

Результат	Система запропонує вибрати кількість гостей, що відвідають заброньований заклад харчування.
-----------	---

Таблиця 2.10 – Опис прецеденту «Введення контактних даних»

Назва	Введення контактних даних
Актор	Користувач
Опис	Для того щоб здійснити бронювання потрібно вказати актуальну інформацію у разі виникнення питань.
Передумови	Обрати дату та й час бронювання, кількість гостей.
Результат	Система проінформує користувача про поточне замовлення та запропонує підтвердити його або скасувати.

Таблиця 2.11 – Опис прецеденту «Вибрати кількість гостей»

Назва	Вибрати кількість гостей
Актор	Користувач
Опис	Надати системі інформацію про кількість відвідувачів закладу харчування, що бронюється.
Передумови	Обрати дату й час для поточного бронювання
Результат	Система запропонує наддасть форму для ведення контактних даних.

Таблиця 2.12– Опис прецеденту «Підтвердити бронювання»

Назва	Підтвердити бронювання
Актор	Користувач
Опис	Підтвердити, що бронювання актуальне і повинно піти у обробку.
Передумови	Надати всі необхідні дані для бронювання.
Результат	Система повідомить користувача про статус замовлення.

2.2 Діаграма варіантів використання для адміністратора

Опишемо варіанти користувача адміністратора. Одними з ключових можливостей адміністратора є дії пов'язані з керуванням інформації про заклад (рис. 2.3).

Керування закладами включає такі дії, як додавання нового закладу до системи (табл. 2.14), редагування інформації про існуючий заклад (табл. 2.15) та видалення інформації про заклад (табл. 2.16).

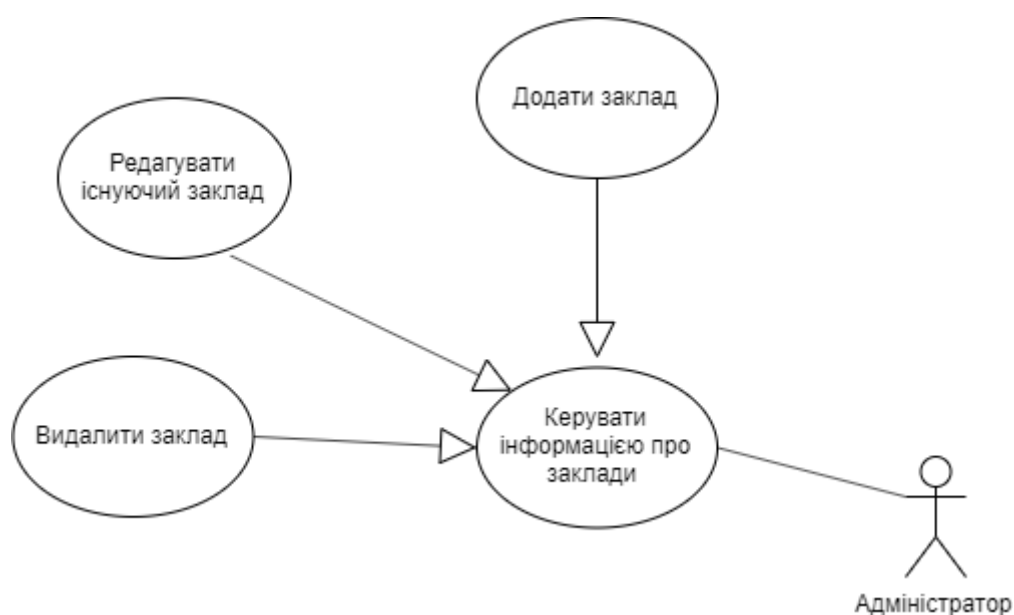


Рисунок 2.3 – Варіанти використання для адміністратора

Таблиця 2.13 – Опис прецеденту «Керувати інформацією про заклад»

Назва	Керувати інформацією про заклад
Актор	Адміністратор
Опис	Надання можливості керувати інформацією про заклад.
Передумови	Авторизуватись на сайті
Результат	Користувачу буде представлений функціонал для керування інформацією та відобразиться таблиця з інформацією про заклад.

Таблиця 2.14 – Опис прецеденту «Додати заклад»

Назва	Додати заклад
Актор	Адміністратор
Опис	Надання можливості додати новий заклад до бази даних
Передумови	Авторизуватись на сайті
Результат	Користувачу буде представлена форма для додання інформацію для створення закладу у системі.

Таблиця 2.15 – Опис прецеденту «Редагувати заклад»

Назва	Редагувати заклад
Актор	Адміністратор
Опис	Редагування інформації вибраного закладу.
Передумови	Авторизуватись на сайті, обрати заклад для редагування.
Результат	Інформація про обраний заклад зміниться в системі.

Таблиця 2.16 – Опис прецеденту «Видалити заклад»

Назва	Видалити заклад
Актор	Адміністратор
Опис	Надання можливості видалити інформацію про заклад з бази даних.
Передумови	Авторизуватись на сайті, обрати заклад.
Результат	Обраний користувачем заклад зникне з бази даних, відобразиться список інших закладів.

Одним з варіантів використання адміністратором є перегляд існуючої статистики (табл. 2.16).

Таблиця 2.16 – Опис прецеденту «Переглянути статистику»

Назва	Переглянути статистику
Актор	Адміністратор
Опис	Переглянути статистику щодо користування ботом, а саме кількість користувачів, частота використання, інформація по регіонам та активність користувачів.
Передумови	Авторизуватись на сайті
Результат	Користувачу будуть представлені таблиці та графіки з відповідною інформацією, що оновлюється в реальному часі.

Висновки до розділу

Були розглянуті функціональні вимоги до системи, а саме для користувача та адміністратора та створена діаграма варіантів використання.

Для кожного варіанта використання описані передумови та результати.

3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Обчислення відстані між закладом харчування та місцезнаходженням користувача

Розглянемо задачу пошуку відстані між двома координатами на карті. Маємо пару координат $c_1(x_1, y_1)$ та $c_2(x_2, y_2)$, де x позначена довгота, а y широта точки на карті, що подані в градусах.

Потрібно знайти відстань d .

Відстані з похибкою 5 градусів (приблизно 550 км) обчислюються за формулою:

$$d^2 = \Delta x^2 + \Delta y^2,$$

де Δx та Δy різниця між широтою та довготою між двома даними точками:

$$\Delta x = x_2 - x_1,$$

$$\Delta y = y_2 - y_1.$$

Отже, відстань розраховується за звичайною формулою пошуку відстані між двома координатами на площині.

3.2 Метод надання рекомендацій на основі контентної фільтрації

Розглянемо метод надання рекомендацій, що базується на контентній фільтрації. Потрібно визначити схожість закладів між собою. Сформуємо масив страв, щоб оцінити схожість за ними. Для кожного слова з меню визначимо значення частоти виникнення або *inverse frequency document* в меню закладів харчування, що обчислюється, за формулою [15]:

$$w_i = \ln \frac{N}{t_i},$$

де t_i – загальна кількість входження даного слова серед усіх меню, а $i = 1, N$.

Щоб присвоїти менше значення загальним словам і приділити більше значення словам, які нечасто зустрічаються частоту виникнення обчислюють оберненим значенням.

Побудуємо матрицю розмірністю $N \times w$, w – загальна кількість слів, що зустрічаються у тексті:

$$\begin{pmatrix} a_{11} & \cdots & a_{1w} \\ \vdots & \ddots & \vdots \\ a_{N1} & \cdots & a_{Nw} \end{pmatrix}.$$

Кожний елемент матриці містить значення частоти появи слова у меню для кожного з закладів харчування.

Розрахунок подібності конкретних об'єктів є поширеною задачею для багатьох методів, що застосовуються при різних проблемах обробки природніх мов та багатьох інших областях. Щоб порівняти схожість двох закладів харчування між собою використаємо формулу косинусу подібності:

$$\text{similarity}(C, D) = \frac{\sum_{i=1}^N C_i * D_i}{\sqrt{\sum_{i=1}^N C_i^2} * \sqrt{\sum_{i=1}^N D_i^2}}.$$

За даною формулою будемо порівнювати кожен з закладів між собою. В результаті маємо матрицю, що буде визначати схожість закладів між собою.

Приклад надання рекомендацій

Нехай маємо такі початкові умови.

Кількість закладів харчування, що порівнюються дорівнює 5.

Кожен з закладів харчування має масив розпізнаних слів, що були взяті з меню. Попередньо виключимо всі некоректні символи та чисельні значення з розпізнаних результатів, щоб полегшити процес створення матриці та збільшити достовірність побудови матриці (таблиця 3.1).

Таблиця 3.1 – Набір даних для аналізу

Назва закладу	Меню
Place1	BACON, OMELET, SALAD, TEA, LINGONBERRY
Place2	CAKE SALAD TUNA CREAM
Place3	TUNA TEA PANCAKE EGGS
Place4	EGGPLANT PANCAKE CHEESE SALAD LATE AMERICANO
Place5	RASPBERRY TUNA SALAD BISCUIT SAUCE

Для подальшої обробки потрібно об'єднати всі меню для та вилучити повтори слів. Розв'яжемо дану задачу аналітично. Створимо матрицю, яка відображає наявність кожного продукту для кожного закладу харчування (рисунок 3.1).

	Place1	Place2	Place3	Place4	Place5
CHEESE			1		
SALAD	1	1		1	1
BISCUIT					1
PANCAKE			1	1	
RASPBERRY					1
AMERICANO				1	
SAUCE					1
TEA	1		1		
LINGONBERRY	1				
CAKE		1			
LATE				1	
TUNA		1	1		1
BACON	1				1
OMELET	1				
EGGS			1	1	
CREAM		1			

Рисунок 3.1 – Відображення наявності у слів в кожному із меню

Нехай користувачу сподобався «Place2», тому що йому сподобалась кухня в другому закладі. користувачу системи хоче підібрати схожий заклад. Аналізуючи

страви, що наявні (таблиця 3.2) у інших закладах, можна зробити такий висновок, що найбільш схожий за стравами заклад «Place5».

Таблиця 3.2 – Порівняння страв між «Place2» та іншими закладами

Місце	Кількість спільних слів	Слова
Place1	1	SALAD
Place3	1	TUNA
Place4	1	SALAD
Place5	2	TUNA, SALAD

Вирішимо дану задачу за допомогою алгоритму.

1) Розрахуємо для кожного закладу матрицю частоти (рисунок 1.2).

[[0 1 0 0 0 0 0 0 0 1 1 0 0 1 0 1 0] [0 0 0 1 0 1 0 0 0 0 0 0 0 1 0 0 1] [0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 1] [1 0 0 0 1 0 1 0 1 0 0 1 0 1 0 0 0] [0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 0 1]]										
	americano	bacon	biscuit	cake	...	salad	sauce	tea	tuna	
Place1	0	1	0	0	...	1	0	1	0	
Place2	0	0	0	1	...	1	0	0	1	
Place3	0	0	0	0	...	0	0	1	1	
Place4	1	0	0	0	...	1	0	0	0	
Place5	0	0	1	0	...	1	1	0	1	

Рисунок 3.2 – Відображення наявності у слів в кожному із меню

2) Визначимо для кожного унікального слова значення IDF (рисунок 2.3)

```
[5 rows x 17 columns]
```

	idf_weights
americano	2.609438
bacon	2.609438
biscuit	2.609438
cake	2.609438
cheese	2.609438
cream	2.609438
eggplant	2.609438
eggs	2.609438
late	2.609438
lingonberry	2.609438
omelet	2.609438
pancake	1.916291
raspberry	2.609438
salad	1.223144
sauce	2.609438
tea	1.916291
tuna	1.510826

Рисунок 3.2 – Відображення наявності у слів в кожному із меню

3) Розрахуємо матрицю подібності, що буде відображати залежність одного закладу від іншого (рисунок 3.3). Чим більше значення елемент матриці має, тим більше заклади схожі між собою. Значення 1 означає, що порівнюється між собою один й той самий заклад, або система має два різних заклади харчування з однаковим меню. Значення 0 означає, що нічого спільного в даних закладах харчування, які порівнюється, немає.

```
[[1.          0.2236068  0.2236068  0.18257419 0.2          ]
 [0.2236068  1.          0.25         0.20412415 0.4472136   ]
 [0.2236068  0.25         1.          0.20412415 0.2236068   ]
 [0.18257419 0.20412415 0.20412415 1.          0.18257419]
 [0.2         0.4472136  0.2236068  0.18257419 1.          ]]
```

Рисунок 3.3 – Матриця подібності для заданого набору

Отже, аналізуючи дані матриці подібності можна зробити висновок, що найбільш схожий заклад харчування до «Place2» є заклад харчування «Place5», а

значення в матриці дорівнює 0.4472. Даний результат співпадає з аналітичними результатом.

Висновок до розділу

У даному розділі був розглянутий метод для процесу підбору закладів харчування та для обчислення відстані між користувачем та закладом харчування. Наведений приклад для підбору рекомендацій.

4 АРХІТЕКТУРА ТА ПРОЕКТУВАННЯ СИСТЕМИ

4.1 Засоби розроблення

Мовою розробки був обраний Python – об'єктно-орієнтована мова програмування, яка набула популярності завдяки чіткому синтаксису [22]. Вихідний код є у вільному доступі та відкритий для модифікації та повторного використання. Python має два фреймворки Flask та Django.

Для інтерфейсу чат-боту Flask – це фреймворк, що надає інструменти, бібліотеки та технології, придатні для створення веб-додатків. Flask надає розробникам різноманітний вибір при розробці веб-додатків, він надає інструменти, бібліотеки та механіку, які дозволяють вам створити веб-додаток. Даний фреймворк буде використовуватися для створення інтерфейсу чат-бота.

Бібліотека Werkzeug для утиліт, призначена для використання з мовою Python. Переважно, це інтерфейс шлюзу веб-сервера або додаток WSGI, який може створювати програмні елементи для функцій запиту, відповіді або утиліти та це інструмент для створення шаблонів для Python [22].

Для серверу додатків використовується Django – веб-фреймворк Python, який дозволяє швидко розроблювати веб-сайти. Він безкоштовний і з відкритим кодом, має процвітаючу та активну спільноту, чудову документацію та багато варіантів безкоштовної та платної підтримки.

Nginx – це веб-сервер HTTP з відкритим кодом та зворотній проксі-сервер. Інструмент став третім з найпопулярніших веб-серверів [26]. PostgreSQL - це система управління реляційними базами даних з, розроблена всесвітньою командою волонтерів. Це база даних з відкритим вихідним кодом, серед усіх конкурентів саме PostgreSQL надає найбільші можливості для розробки [25]. Всі помилки та труднощі пов'язані з використанням даного інструменту обговорюються у спільнотах користувачів. Дана СУБД має такі переваги:

- можливість підключатися до декількох серверів, ядер та користувачів одночасно;
- необмежна можливість копіювання та розповсюдження;

- самостійне доповнення функцій за потреби.

Для того, щоб підібрати заклади, які задовільнять користувача потрібно вирішити проблему обробки меню. Зазвичай, щоб знайти меню люди користуються пошуковими системами, щоб знайти сайт про заклад та меню. Простіше та й швидше було би знайти заклади по стравам.

У системі всі меню збираються вручну, коли в закладі оновлюється меню заклад через форму зворотнього зв'язку може надіслати його. Для того щоб розпізнавати текст система використовує Google OCR – це безкоштовний, але лімітований інструмент [27]. Комунікація здійснюється за допомогою інтерфейсу, що розпізнає текст незалежно від мови користувача.

4.2 Опис процесу діяльності

Основні процеси, що відбуваються у чат-боті – процес пошуку закладів харчування та процес бронювання. Розглянемо ці процес згідно з структурни схем діяльності, що наведена в додатках.

Щоб знайти заклад харчування потрібно визначити яким чином буде здійснюватися пошук: за текстовим запитом (позиція меню, вулиця, назва закладу), чи місцезнаходженням. Якщо обрано пошук за запитом, існує можливість відсортувати результати пошуку за місцезнаходженнями.

Далі при вдалому пошуковому запиті відбувається процес отримання результатів та перегляду інформації про результати пошуку. У разі коли результатів недостатньо існує можливість збільшити кількість результатів. Якщо було вибрано конкретний заклад харчування, то є можливість забронювати заклад.

Процес бронювання закладів харчування складається з послідовний дій, що включають надсилання дати, часу, кількості гостей та номер телефону. Потім потрібно підтвердити бронювання. У разі підтвердження бронювання, його можна скасувати у будь-який час. Структурна схема діяльності для процесу бронювання розміщена у додатках.

4.3 Проектування бази даних

Для проектування бази даних системи визначимо сутності, що будуть основою для таблиць бази даних.

Сутність «Заклади» включає основні властивості закладів харчування (таблиця 4.1). Саме ці дані слугують основним джерелом для здійснення пошуку закладів в системі.

Таблиця 4.1 – Таблиця «Places»

Поле	Опис	Тип даних	Первинний ключ	Зовнішній ключ
PLACE_ID	Ідентифікатор закладу	INT	X	
PLACE_NAME	Назва закладу	VARCHAR		
ADDRESS	Адреса	VARCHAR		
PHONE	Телефон	VARCHAR		
LONGITUDE	Широта	DOUBLE		
LATITUDE	Довгота	DOUBLE		
PRIORITY	Статус закладу	INT		
PRICE	Цінова категорія	INT		
LINKS_GM	Посилання на карти	VARCHAR		
MENU	Перелік позицій меню	JSON		
CATEGORY	Команда для пошуку	VARCHAR		
VIEW_LINK	Посилання на інформацію про заклад	VARCHAR		

Було створено сутність «Розклад» (таблиця 3.2), яка містить властивості пов'язані з робочими годинами кожного дня тижня для кожного закладу з сутності «Заклади».

Таблиця 4.2 – Таблиця «Schedule»

Поле	Опис	Тип даних	Первинний ключ	Зовнішній ключ
ID	Ідентифікатор закладу	INT	X	
PLACE_ID	Ідентифікатор закладу	VARCHAR		X
DAY	День тижня	INT		
W_FROM	Час відкриття	TIME		
W_TO	Час закриття	TIME		

«Команди» теж є окремою сутністю (таблиця 4.3), що описує дії користувача для подальшого аналізу. До дій користувача відноситься вся активність в системі, а саме інформація про користувача та заклади, які йому пропонуються, дату та час здійсненої користувачем дії, реакцію системи на запити користувача.

Таблиця 4.3 – Таблиця «Log»

Код	Опис	Тип даних	Первинний ключ	Зовнішній ключ
RECORD_ID	Ідентифікатор запису	INT	X	
FIRST_NAME	Ім'я користувача	VARCHAR		
LAST_NAME	Прізвище користувача	VARCHAR		
USERNAME	Посилання на акаунт	VARCHAR		
LONGITUDE	Широта	DOUBLE		
LATITUDE	Довгота	DOUBLE		
DATE	Дата виконання команди	DATETIME		
COMMAND	Вид команди	PRICE		

Код	Опис	Тип даних	Первинний ключ	Зовнішній ключ
FUNC_BOT	Текст команди	VARCHAR		
PLACE_ID	Ідентифікатор закладу	JSON		X
PRIORITY	Статус закладу	VARCHAR		
TIME	Час виконання команди	VARCHAR		
USER_ID	Ідентифікатор користувача	VARCHAR		X
LINK	Посилання на заклад	VARCHAR		

Щоб зафіксувати активність користування системою було створено сутність «Частота» (таблиця 4.4).

Таблиця 4.4 – Таблиця «UserUsage»

Назва	Опис	Тип даних	Первинний ключ	Зовнішній ключ
RECORD_ID	Ідентифікатор запису	INT	X	
DAY_OF_USAGE	Кількість днів використання за весь час	INT		
START_DATE	Дата першого використання	DATETIME	X	
END_DATE	Дата останнього використання	DATETIME		
LAST_MONTH_USAGE	Кількість днів використання за останній місяць	INT		

Назва	Опис	Тип даних	Первинний ключ	Зовнішній ключ
USER_ID	Ідентифікатор користувача	VARCHAR		X
START_DAY	Перший день використання	INT		

Сутність «Кількість користувачів» (таблиця 4.5) описує дані про новоприбулих користувачів за добу, що опрацьовуються функцією статистики.

Таблиця 4.5 – Таблиця «UserAmount»

Код	Опис	Тип даних	Первинний	Зовнішній ключ
ID	День роботи	INT	X	
USER_ID	Ідентифікатор користувача	VARCHAR		X
NEW_USERS_BY_DAY	Кількість нових користувачів за день	VARCHAR		
USER_AMOUNT	Кількість користувачів	VARCHAR		
BLOCKED	Кількість заблокованих	DOUBLE		
DAY_OF_WEEK	День тижня	DOUBLE		
NUMBER_OF_WEEK	Номер тижня	INT		

При пошуку закладів формується масив результатів для кожного з користувачів, щоби не проводити пошук велику кількість разів було визначено сутність «Сесія» (таблиця 4.6).

Таблиця 4.6 – Таблиця «Sessions»

Код	Опис	Тип даних	Первинний ключ	Зовнішній ключ
RECORD_ID	Ідентифікатор запису	INT	X	
LONGITUDE	Широта	DOUBLE		
LATITUDE	Довгота	DOUBLE		
DATE	Остання дата сесії	DATETIME		
PLACE_LIST	Результати пошуку за запитам	JSON		
TEXT	Останній текстовий запит	VARCHAR		
STATUS	Вид запиту	VARCHAR		
USER_ID	Ідентифікатор користувача	VARCHAR		X
MESSAGE_ID	Ідентифікатор запиту	INT		

Оскільки інструменти платформ обміну миттєвими повідомленнями не зберігають ідентифікатори про активних та заблокованих користувачів бота є потреба зафіксувати цю інформацію у базі даних самостійно. Сутність «Користувачі» описує активність користувачів (таблиця 3.7).

Таблиця 4.7 – Таблиця «Users»

Поле	Опис	Тип даних	Первинний ключ	Зовнішній ключ
USER_ID	Ідентифікатор закладу	VARCHAR	X	

Поле	Опис	Тип даних	Первинний ключ	Зовнішній ключ
BLOCKED	Статус активності користувача	BOOLEAN		

Сутність «Бронювання» (таблиця 4.8) описує процес бронювання закладів, та має такі властивості : номер бронювання, заклад, що бронюється, кількість осіб, дату створення, дату бронювання

Таблиця 4.8 – Таблиця «Booking»

Поле	Опис	Тип даних	Первинний ключ	Зовнішній ключ
ID	Ідентифікатор бронювання	INT	X	
PLACE_ID	Ідентифікатор закладу	VARCHAR		X
PLACE_INFO	Інформація про заклад, що заброньований	JSON		
DATE	Час створення	TIME		
AMOUNT	Кількість відвідувачів	INT		
STATUS_FOR_OPERATOR	Статус бронювання	VARCHAR		
TIME_OF_BOOKING	Час бронювання	DATETIME		

Структурна схема бази даних наведена у додатку.

4.4 Попередня обробка текстових даних

Користувачі надсилають у бот текстові запити, що можуть містити багато помилок, які призведуть до неточного підбору закладів. Наведемо помилки, які зустрічається при надсиланні запитів у таблиці 4.9.

Таблиця 4.9 – Типові помилки у текстових запитах

Помилки в запитах	Опис проблеми	Методи рішення
Мова запиту	Оскільки системою користуються не тільки україномовні користувачі, то існує ймовірність запитів з інших мов.	Переклад текстових запитів на мову, що використовується у меню в закладі.
Лематизація	Користувач не завжди надсилає запит в називному відмінку у множині, що важливо для пошуку з меню ресторанів.	Підключити відповідні сервіси, щоб розпізнати відмінок слова.
Кількість слів	Запит може складатися з декількох слів у різних відмінках.	Зробити розбір запитів, розбити запит на масив провести аналіз окремо по слову.
Синоніми та помилки написання	Назви страв мають декілька синонімів, але зазвичай у меню вказують тільки один варіант запитів.	Включити словник синонімів для загальноновживаних страв.

Помилки написання	Багато страв походять від іншомовних слів і тому існує проблема, коли користувачі роблять помилки в написанні страв у запитах (напр. хумус, пишуть хуммус).	Створити словник з виключеннями для помилок.
Уживання великої та малої літер	Текстові запити можуть містити різні літери, що ускладнює процес пошуку.	Уніфікація запитів до одного виду літер.

Всі рішення проблем при формуванні текстових запитах спростять обробку запитів у чат-боті та реалізовані у сервері додатків.

4.5 Архітектура програмного забезпечення

Система має клієнт-серверну архітектуру. На рисунку 4.1 представлена структурна схема системи.

Розглянемо основні елементи системи. Головним компонентом є сервер додатків, який включає функції, що пов'язані з наданням рекомендацій, обробкою меню та статистикою та перетворенням даних.

Чат-бот – веб-сервер, що опрацьовує всі запити від користувача та надсилає їх через API для подальшого перетворення. Веб-сервер слугує для відображення інформації про статистику для адміністратора системи та включає можливість керувати інформацією про заклади. Всі складові програми розташовані на одному сервері та взаємодіють між собою локально. Доступ до нього надається за допомогою Proxy Server. Це сервер, що дає доступ користувачам та адміністратору до системи.

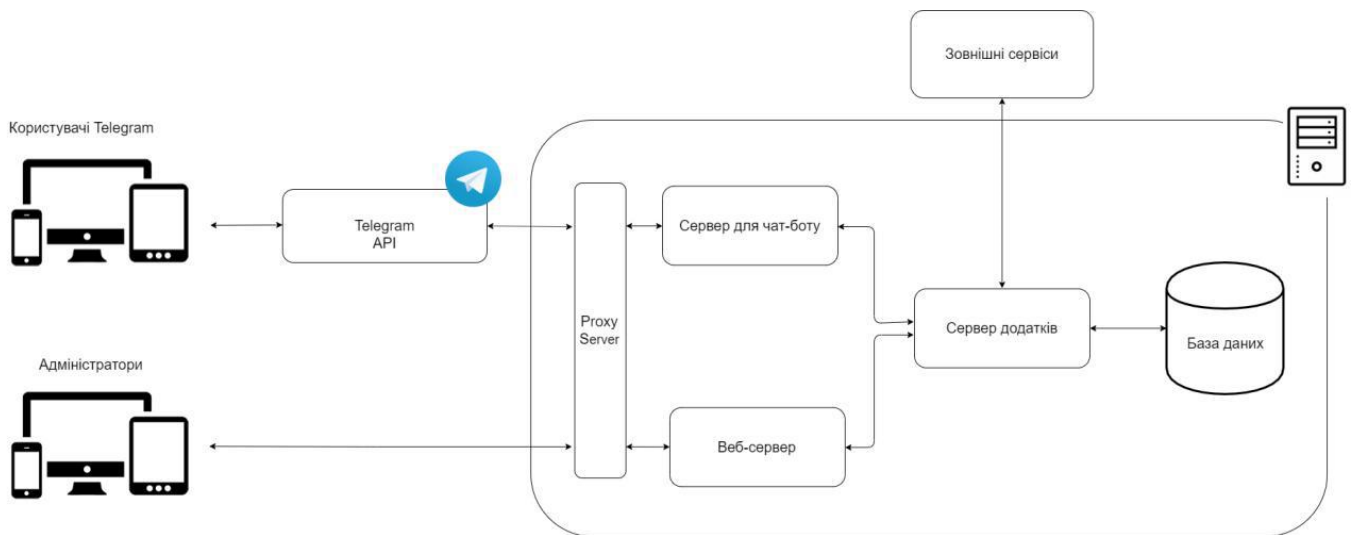


Рисунок 4.1 – Структурна схема системи

Доступ до чат-бота попередньо здійснюється за допомогою Telegram API. Для того, щоб створити чат-бота необхідно здійснити декілька кроків.

1. Створити акаунт в Telegram.
2. Створити чат-бота за допомогою спеціального сервісу.
3. Вказати назву, посилання для нового чат-бота.
4. Отримати спеціальний ключ для доступу.

Завершуючи дані етапи необхідно створити програму, що буде опрацьовувати запити користувача та зворотній зв'язок між ботом та Telegram API. На вхід будуть прийматися запити від користувачів бота та оброблятися згідно з існуючою логікою.

Існують два варіанти взаємодії між сервером та Telegram API: WebHook та LongPolling. Webhook – це механізм оповіщення системи про події, що здійснив користувач [11]. Розробник визначає адресу на які Telegram буде відправляти всі запити від користувача. За визначеною адресою розташований сервер з логікою для чат-бота (рисунок 4.2).

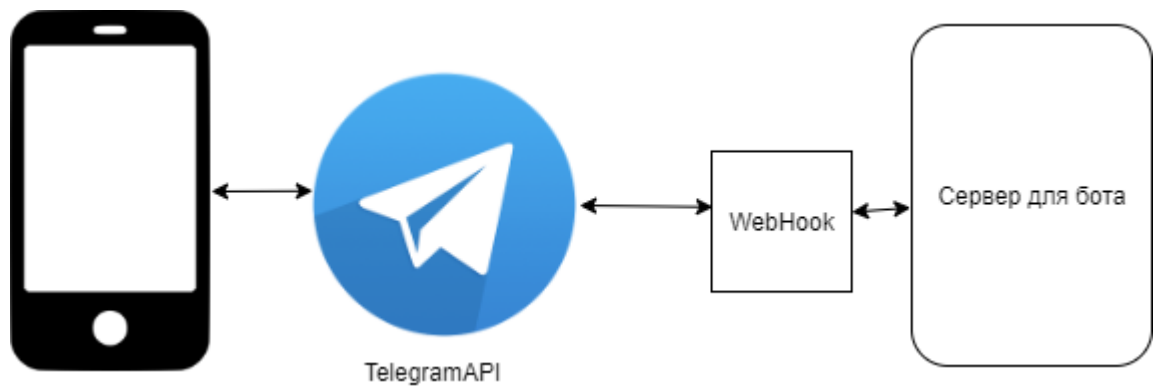


Рисунок 4.2 – Схема роботи Webhook

У додатках представлена структурна схема розгортання для чат-боту. Користувачу для взаємодії з чат-ботом потрібно встановити додаток Telegram через якого відбувається взаємодія з ботом. Коли користувач відправляє запит, Telegram API надсилає його до серверу чат-боту, який оброблює надісланий текстовий запит та надсилає до серверу з логікою. Дочекавшись відповіді від API, веб-сервер відправляє відповідь користувачу. Схема структурна компонентів описує пристрої та взаємодії між ними для побудови чат-бота. На одному сервері розташовано база даних, сервер для логіки та веб-сервер для чат-боту. Сервер для чат-ботом взаємодіє з Telegram для обміном запитів між пристроєм користувача та рекомендаційною системою.

Головним компонентом системи є сервер додатків, який виконує такі функції:

- надає рекомендації;
- оброблює запити;
- оновлює дані про меню;
- взаємодіє з Google OCR.

4.6 Діаграма послідовності для процесу пошуку

Пошук закладів харчування здійснюється у такій послідовності. Користувач надсилає запит, йому повертаються, результати пошуку, які він переглядає. Якщо користувача не задовольняють результати пошуку, він переглядає інші результати надіславши запит. Для того, щоб показати як взаємодіють частини системи між собою у додатках наведена схема структурна послідовності.

Об'єктами системи є LogicServer – виконує основні функції системи, чат-бот інтерпретує та надсилає дані від користувача, база даних, що реагує на команди LogicServer. TelegramAPI виступає інтерфейсом для взаємодії з користувачем.

Перед тим як надавати рекомендації користувач відправляє запит в додаток Telegram, який перенаправляє їх до ChatBot через API. Дані від користувача тексту та передаються до Logic Server, який за допомогою бази даних та відповідних функцій опрацьовує запит, через функції попередньої обробки тексту та надає результати у вигляді рекомендацій. Результати повертаються до користувача, за допомогою інтерфейсу чат-бота. Оскільки, не всі дані одразу надсилаються користувачу тому, інші результати пошуку зберігаються до бази даних.

Опрацювання додаткових результатів для користувача відбуваються у тій же послідовності, як для пошуку закладів харчування, але в даному процесі запит не оброблюється у Logic Server, а повертається вже готовим результатом до користувача попередньо збереженим у базі даних.

4.7 Розроблення інтерфейсу

Процес розроблення інтерфейсу поділяється на два етапи. Оскільки основна частина системи розробляється у вигляді чат-боту, то створення інтерфейсу для пошуку закладів харчування буде створено за допомогою інструментів платформи Telegram.

Для того, щоб розпочати роботу з ботом необхідно натиснути команду “/start”, після чого з'явиться головне меню з відповідними кнопками (рис. 4.1).

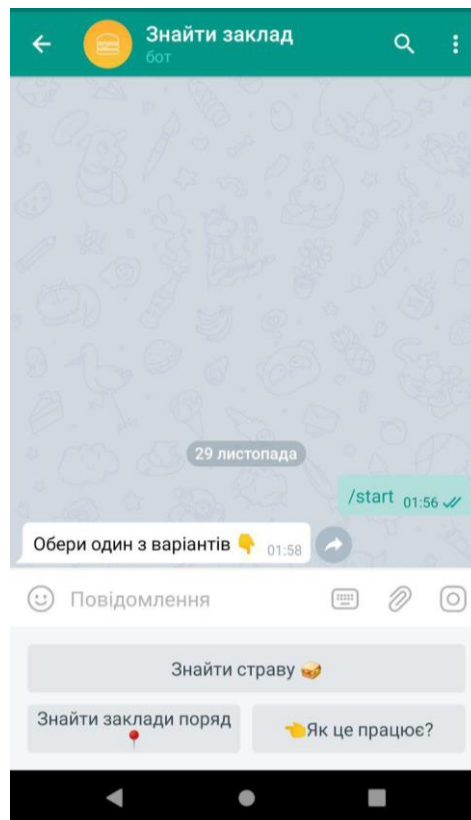


Рисунок 4.3 – Головне меню

Почати процес пошуку закладу можна за допомогою кнопки «Знайти заклади» або «Знайти заклади поряд», що дозволить шукати заклади по текстовим запитам або по місцезнаходженню відповідно.

Для того, щоб відправити запит потрібно відправити запит у форму для відправки повідомлення, що зображена на рисунку 4.2. Після введення текстового запиту бот відправляє повідомлення про кількість результатів пошуку та пропонує показати всі результати або варіанти відсортовані за місцезнаходженням.



Рисунок 4.4 – Пошук за текстовим запитом

Наступним кроком є перегляд результатів, користувачу пропонуються по два результати, що підібрала система (рис.4.4)

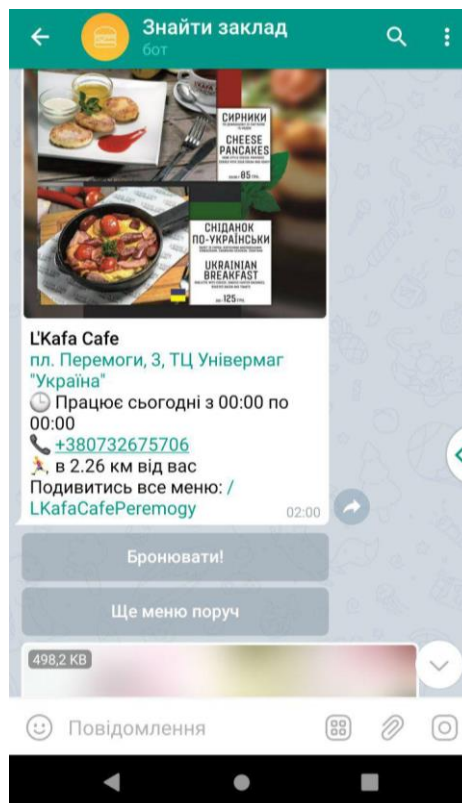


Рисунок 4.4 – Результати пошуку

Спочатку пропонуються тільки варіанти в яких є відкритими, а потім закритими. Користувач може переглянути інформацію про заклад подивитися меню, контактні дані та забронювати заклад.

У випадку коли за запитом відсутні результати або коли запит некоректний система рекомендує заклади (рис 4.6)

Детальний перегляд інформації про ресторан потребує створення веб-сторінки із-за обсягу інформації. В Telegram існує гнучкий інструмент для впровадження веб сторінок InstantView, що дозволяє користувачам Telegram переглядати дані сторінки з веб-сторінок з мінімальним часом завантаження. Всі статі доступні за посиланням та адаптуються до формату додатку. За допомогою системи миттєвого перегляду користувачі Telegram можуть читати статі з будь-яких засобів масової інформації чи з блогів уніфікованим способом. Приклад застосування даного інструменту зображений на рисунках 4.7 та 4.8.

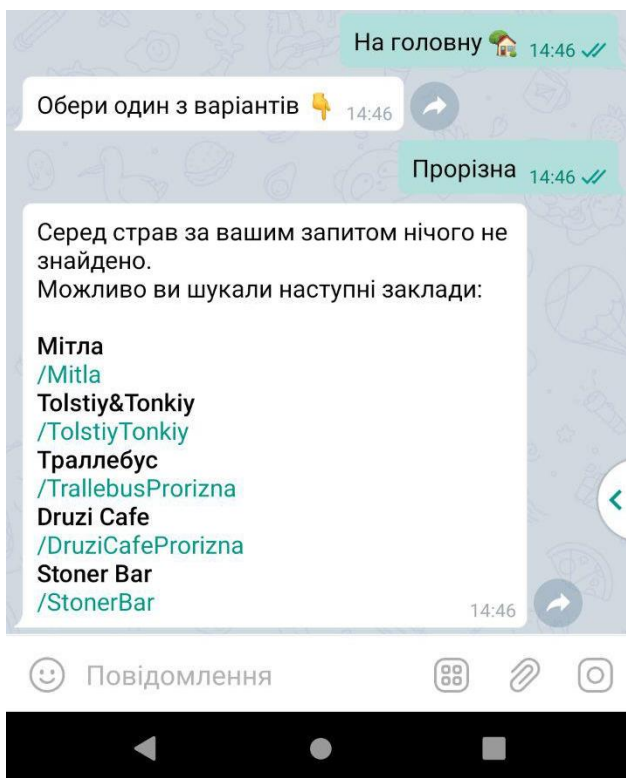


Рисунок 4.6 – Рекомендації щодо закладів

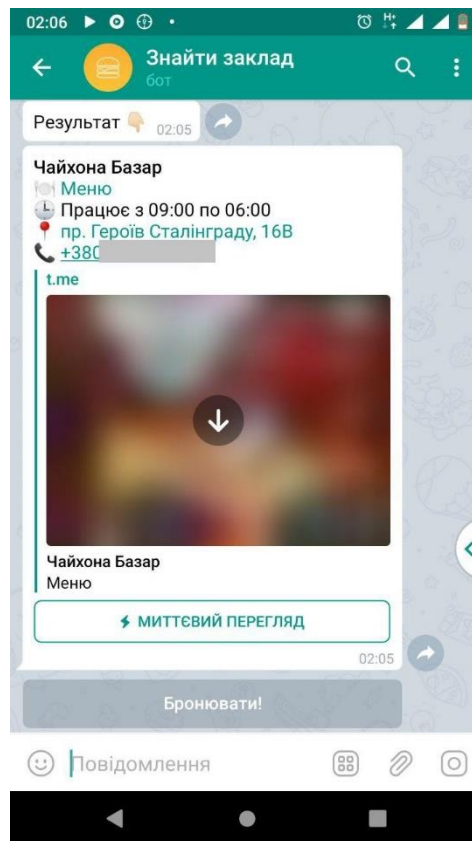


Рисунок 4.7 – Перегляд інформації про заклад

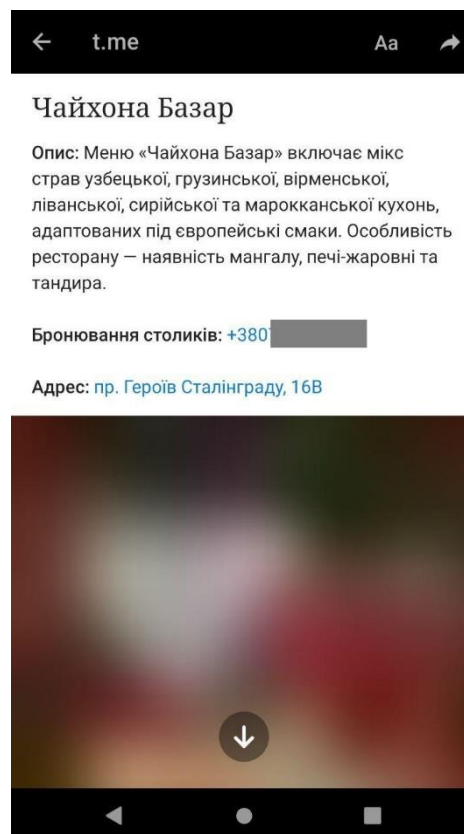


Рисунок 4.8 – Детальний перегляд інформації про заклад

Якщо виникає потреба забронювати обраний заклад, користувач повинен натиснути кнопку «Бронювати» у інтерфейсі та заповнити такі поля за допомогою текстових запитів. До даної інформації включено такі параметри:

- дата бронювання;
- час бронювання;
- кількість гостей;
- ім'я на кого бронюється заклад;
- контактний телефон.

При вдалому процесі бронювання користувач отримає відповідне повідомлення з відповіддю про бронювання. Сам процес бронювання зображений на рисунках 4.9.

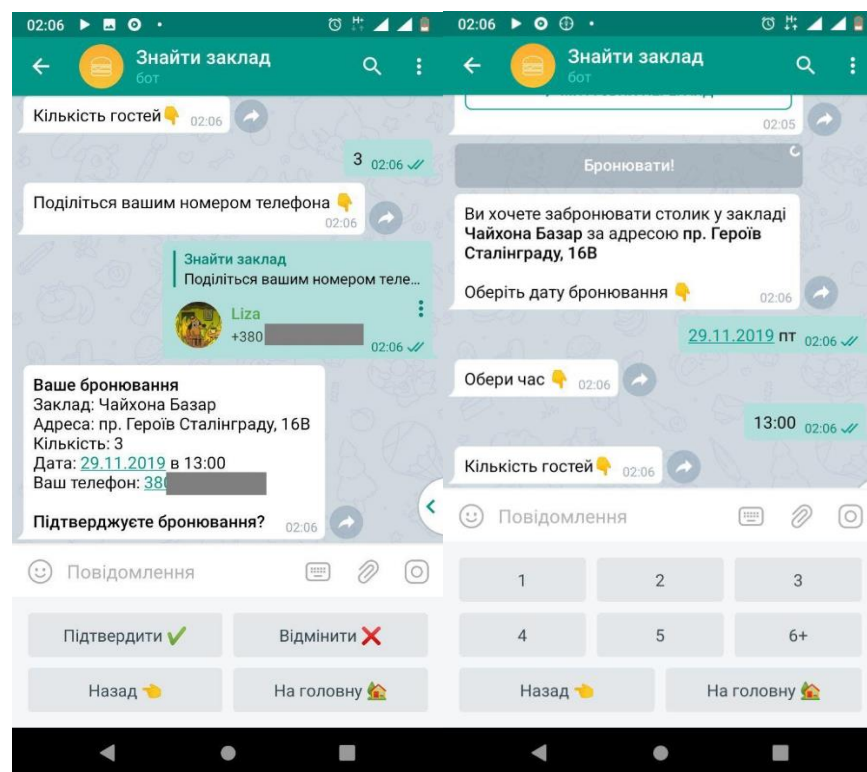


Рисунок 4.9 – Процес бронювання

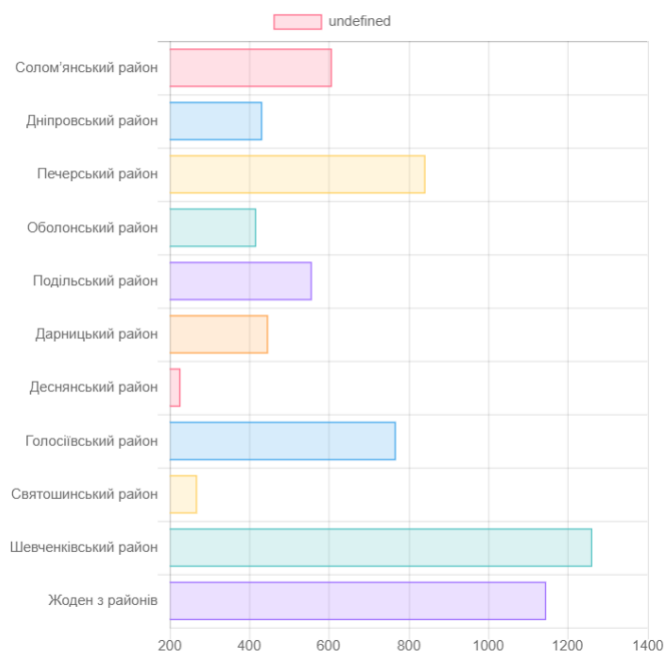
Для того, щоб спостерігати за статистикою використання та здійснювати контроль за інформацією буде створений веб-сайт за допомогою фреймворку Django.

Інтерфейс для модулю статистики включає в себе інформацію про частоту використання користувачами для кожного з району міста та окремо виділяючи

інформацію про запити, що залежать тільки від місцезнаходження користувача. Також фіксується частота використання по регіонам (рисунок 4.10).

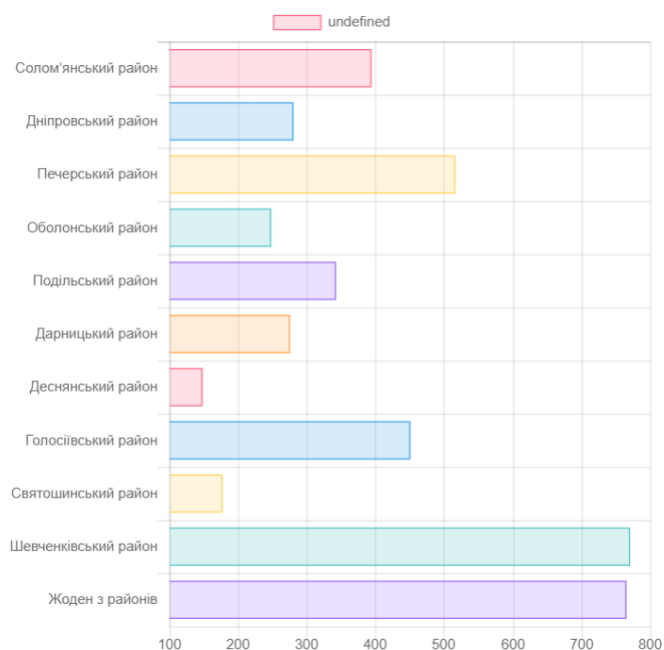
Частота використання по регіонам

Всі запити



Район	%
Солом'янський район	8
Дніпровський район	6
Печерський район	12
Оболонський район	5
Подільський район	7
Дарницький район	6
Деснянський район	3
Голосіївський район	11
Святошинський район	3
Шевченківський район	18
Жоден з районів	16

Запити на пошук закладів поруч



Район	%
Солом'янський район	9
Дніпровський район	6
Печерський район	11
Оболонський район	5
Подільський район	7
Дарницький район	6
Деснянський район	3
Голосіївський район	10
Святошинський район	4
Шевченківський район	17
Жоден з районів	17

Рисунок 4.10 – Сторінка інформації про запити по регіонам

Для того, щоб відстежувати нових користувачів була створена сторінка, що містить інформацію про кількість користувачів, заблокованих користувачів, кількість нових користувачів. На рисунку 4.11 зображено відповідні таблиці з даними та графіки, що фіксують дані кожного дня.

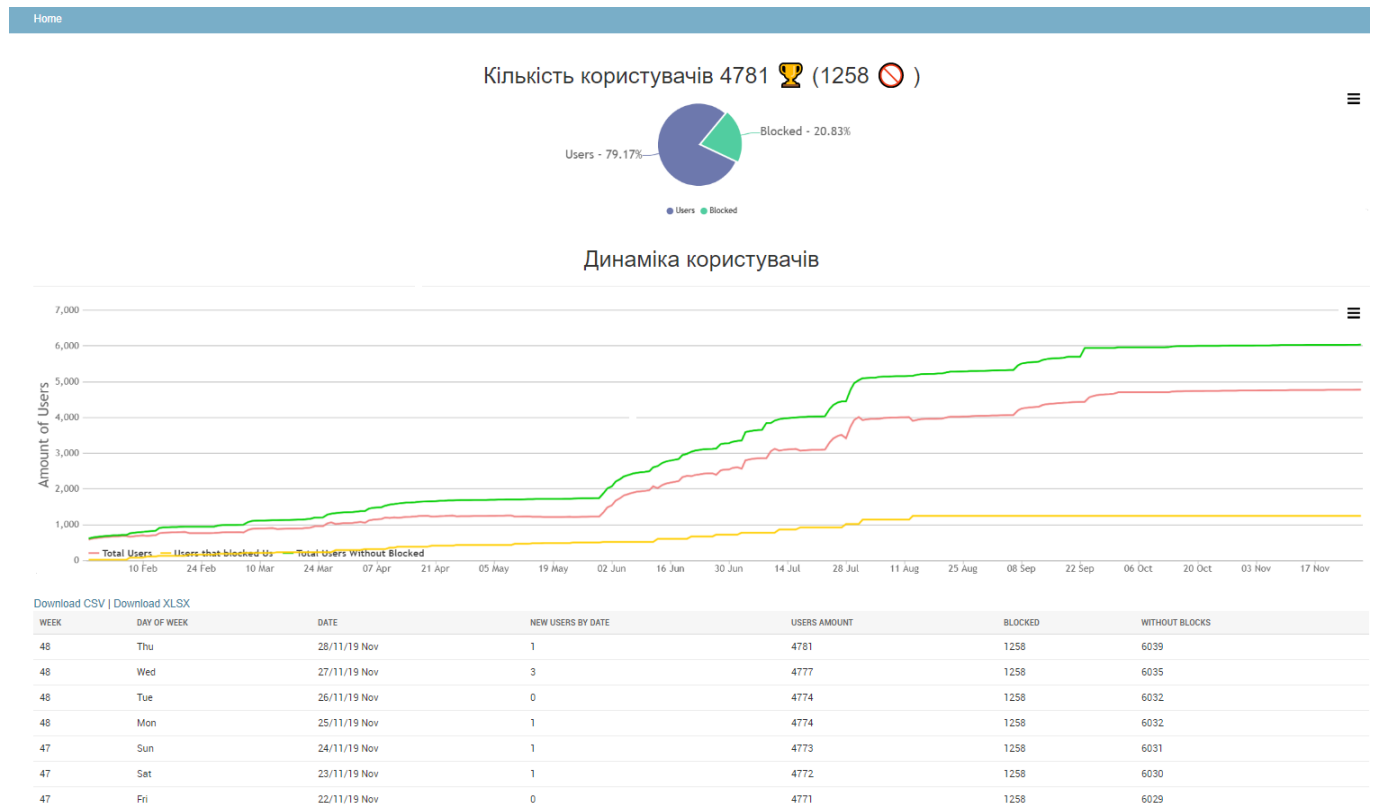


Рисунок 4.11 – Сторінка з інформацією про кількість користувачів

Частота використання чат-боту є сторінкою для відображення інформації про користувачів так їх активність (рис. 4.12)

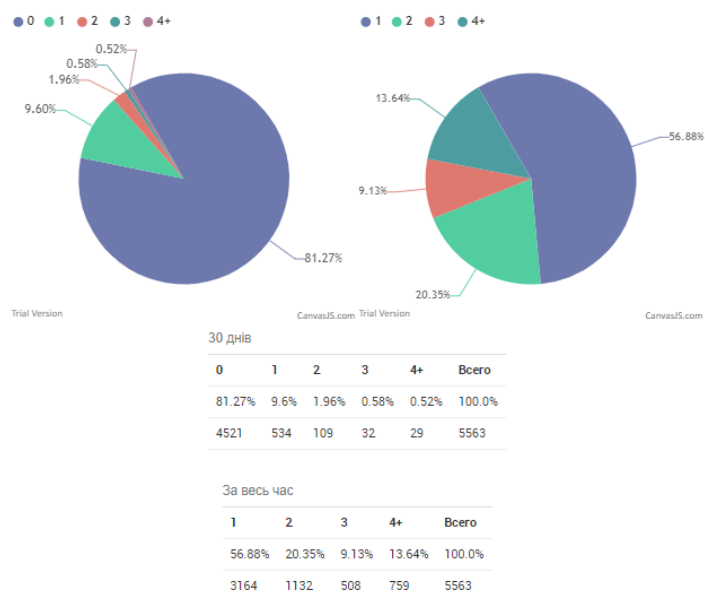
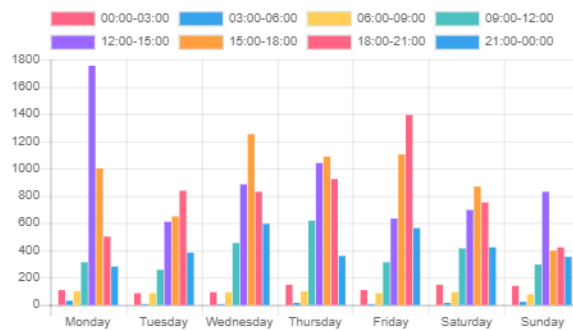


Рисунок 4.12 – Сторінка з інформацією про частоту використання

Відстеження активності запитів для кожного дня тижня та проміжку часу фіксується на сторінці фактор голоду (рис. 4.13).

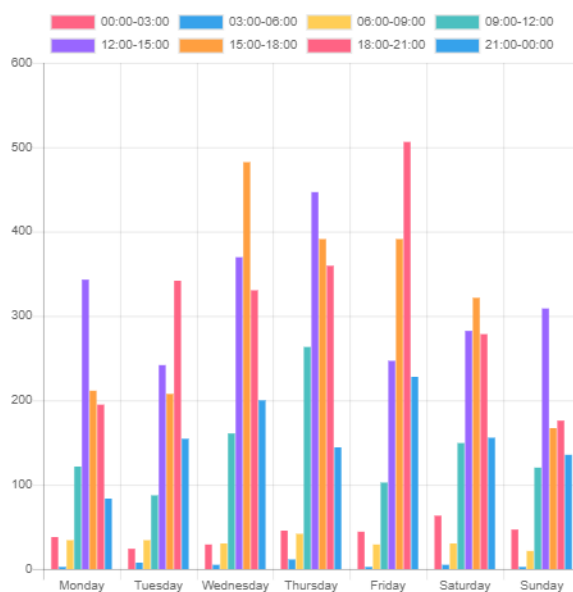
Фактор голоду

Всі запити



time	Mon	Tue	Wed	Thu	Fri	Sat	Sun
00:00-03:00	110	84	92	150	106	147	144
03:00-06:00	30	11	9	19	10	15	22
06:00-09:00	103	85	94	98	87	97	76
09:00-12:00	313	261	456	622	311	418	300
12:00-15:00	1754	611	884	1044	631	700	826
15:00-18:00	1004	650	1249	1091	1101	866	402
18:00-21:00	498	836	831	926	1392	751	426
21:00-00:00	281	383	598	360	561	425	355

Пошук закладів



time	Mon	Tue	Wed	Thu	Fri	Sat	Sun
00:00-03:00	38	24	29	45	44	63	47
03:00-06:00	2	7	5	12	3	5	2
06:00-09:00	34	34	31	42	29	31	21
09:00-12:00	122	87	161	263	102	150	120
12:00-15:00	343	242	370	447	247	282	309
15:00-18:00	211	207	482	391	391	321	167
18:00-21:00	195	342	330	359	506	278	176
21:00-00:00	83	155	200	144	228	156	136

Рисунок 4.13 – Сторінка з інформацією про фактор голоду

Адміністратор системи може керувати інформацією про заклад перед тим як авторизується в формі, що зображена на рисунку 4.14. Редагування інформації про заклад включає форму, що містить всі деталі про заклад (рис.4.15).

Вхід до системи пошуку закладів

Логін

Пароль

Увійти

Рисунок 4.14 – Сторінка з інформацією про частоту використання

Змінити інформацію про заклад

Home · Api · Offers · Offers object (847508)

Змінити інформацію

Place name: CATCH

Place id: 10101309

Address: вул. Володимирська, 12

Phone:

Longitude: 30,5170669

Latitude: 50,4558505

Text: Рибний ресторан з інтер'єром, що поєднує

Text id: 10101309

Price:

W from: 11:00:00 Now | ⌚

W to: 00:00:00 Now | ⌚

Day: 3

Cuisine data:

Рисунок 4.15 – Сторінка з інформацією про частоту використання

Висновок до розділу

У даному розділі було розглянуто процес проектування програмного забезпечення.

Був описаний розглянуто структуру бази даних системи та архітектура програмного продукту на прикладі структурних схем системи.

5 ТЕСТУВАННЯ СИСТЕМИ

Для того, щоб визначити чи всі заявлені вимоги наявні в програмному продукті потрібно протестувати система. Коректність роботи системи перевіряється за допомогою двох видів тестування, а саме ручного та модульного. API програмного продукту тестується за допомогою додатку PostMan.

5.1 Ручне тестування

У таблицях 5.1-5.12 наведені результати ручного тестування. Для кожного з варіанту використання системи наведемо відповідний тест.

Для тестування у ролі користувача тестування повинно проводитися у інтерфейсі платформи обміну миттєвими повідомленнями. Вимоги, що тестуються у ролі адміністратора проводяться на веб-сайті.

Таблиця 5.1 – Тест пошуку закладів

Дія:	Знайти заклад	
	Результат	Статус тесту
Передумова		
Запустити бот	Увійти у інтерфейс бота або написати команду «/start»	успішно
Кроки тесту		
Натиснути на кнопку «Знайти заклади»	З'являється повідомлення про можливість вводу текстового запиту у поле для надсилання повідомлення	успішно
Післяумова		
Ввести запит	Здійснення пошуку	успішно

Таблиця 5.2 – Тест надсилання місцезнаходження

Дія:	Надіслати місцезнаходження	
	Результат	Статус
Передумова		
Натиснути кнопку «Знайти страву»	Надіслане повідомлення про можливість пошуку	успішно
Кроки тесту		
Натиснути на «Надіслати місцезнаходження» та підтвердити надсилання	Від користувача надіслане повідомлення з картами	успішно
Післяумова		
Перегляд результатів	Відображаються результати пошуку	успішно

Таблиця 5.3 – Тест введення запиту

Дія:	Введення запиту	
	Результат	Статус
Передумова		
Відкрити пошук меню	Надіслане повідомлення про можливість шукати	успішно
Кроки тесту		
Вести текстовий запит	Повідомлення про кількість результатів	успішно
Післяумова		
Натисніть на кнопку «На головну»	Відкрита стартова сторінка інтерфейсу	успішно

Таблиця 5.4 – Тест перегляду результатів

Дія:	Перегляд результатів	
	Результат	Статус
Передумова		
Здійснити пошук	Надіслане повідомлення про кількість результатів	успішно
Кроки тесту		
Натиснути «Показати всі» або «Показати найближчі»	Надіслане повідомлення про результат з інформацією про заклад	успішно
Післяумова		
Натисніть на кнопку «На головну»	Відкрита стартова сторінка інтерфейсу	успішно

Таблиця 5.5 – Тест забронювати

Дія:	Забронювати	
	Результат	Статус тесту
Передумова		
Переглянути результати пошуку, обрати заклад для бронювання	Відображаються результати пошуку	успішно
Кроки тесту		
Натиснути на кнопку «Бронювати», ввести дату й час, кількість гостей, телефон	Повідомлення, що повідомляє, що бронювання знаходить в роботі	успішно
Післяумова		
Натисніть на кнопку «На головну»	Відкрита стартова сторінка інтерфейсу	успішно

Таблиця 5.6 – Тест скасування результатів

Дія:	Скасувати результати	
	Результат	Статус тесту
Передумова		
Забронювати будь-який заклад	Існує активне бронювання та кнопка «Скасувати» присутня на панелі кнопок	успішно
Тест		
Натиснути «Скасувати»	Повідомлення, що бронювання відмінено	успішно
Післяумова		
Натиснути на кнопку «На головну»	Відкрита стартова сторінка інтерфейсу	успішно

Таблиця 5.7 – Тест надсилення локації

Дія:	Надіслати локацію	
	Результат	Статус тесту
Передумова		
Натиснути кнопку «Знайти заклад»	Користувачу запропоновано знайти заклад	успішно
Тест		
Натиснути кнопку для відправлення місцезнаходження «Додати розташування»	Користувачу будуть надіслані результати	успішно
Післяумова		
Натиснути на кнопку «На головну»	Відкрита стартова сторінка	успішно

Таблиця 5.8 – Тест керування закладами

Дія:	Керування існуючими закладами	
	Результат	Статус тесту
Передумова		
Авторизуватись	Перехід на сторінку «Головна»	успішно
Тест		
Натиснути на посилання «Заклади»	Відкриється інтерфейс для керування базою з закладами харчування	успішно
Післяумова		
Натиснути на кнопку «На головну»	Відкрита стартова сторінка	успішно

Таблиця 5.9 – Тест додавання закладу

Дія:	Додати заклад	
	Результат	Статус тесту
Передумова		
Авторизуватись та натиснути на посилання «Заклади»	Відкриється інтерфейс для керування базою з закладами харчування	успішно
Тест		
Натиснути на посилання «Додати заклад», заповнити поля для нового закладу, натиснути на кнопку «Зберегти»	Повідомлення про додавання нового закладу	успішно
Післяумова		
Натиснути на посилання «На головну».	Відкрита головна сторінка для адміністратора	успішно

Таблиця 5.10 – Тест на редагування закладу

Дія:	Редагувати заклад	
	Результат	Статус тесту
Передумова		
Авторизуватись та натиснути на посилання «Заклади»	Відкриється інтерфейс для керування базою з закладами харчування.	успішно
Тест		
Натиснути посилання «Редагувати заклад» та відредагувати потрібні поля, натиснути кнопку «Зберегти»	Повідомлення інформація про заклад харчування змінилася.	успішно
Післяумова		
Натиснути на кнопку «На головну»	Відкрита головна сторінка для адміністратора	успішно

Таблиця 5.11 – Тест на видалення закладу

Дія:	Видалити заклад	
	Результат	Статус тесту
Передумова		
Авторизуватись та натиснути на посилання «Заклади»	Відкриється інтерфейс для керування базою з закладами харчування	успішно
Тест		
Натиснути на конкретний заклад та натиснути на кнопку «Видалити»	Повідомлення що заклад видалений з бази даних	успішно
Післяумова		
Натиснути на посилання «На головну»	Відкрита головна сторінка для адміністратора	пройдений

Таблиця 5.12 – Тест на перегляд статистики

Дія:	Переглянути статистику	
	Результат	Статус тесту
Передумова		
Авторизуватись та натиснути на посилання «Заклади»	Відкриється інтерфейс для керування базою з закладами харчування	успішно
Тест		
Натиснути на одне з посилань «Фактор голоду», «Частота користування», «Інформація по регіонами»	Відкриється сторінка з інформацією про заклади	успішно
Післяумова		
Натиснути на посилання «На головну»	Відкрита головна сторінка для адміністратора	успішно

5.2 Модульне тестування

У модульному тестуванні програмне забезпечення ділиться на менші частини (функції, процедури, інтерфейси) для того, щоб перевірити їх придатність до використання. Приведемо переваги даного типу тестування:

- розпізнає помилки завчасно;
- спрощує процес налагодження;
- покращує якість коду.

У мові програмування Python існує підтримка даного виду тестування у вигляді фреймворку unittest. Фреймворк є гнучким інструментом в якому можна створити

тестові набори (наприклад, в окремі класи або в окремі файли). Даний фреймворк використовує методологію JUnit, що написана на мові програмування Java.

Розглянемо детально процес бронювання та напишемо тести, щоб перевірити коректність роботи даної функції.

Процес бронювання вимагає у користувача велику кількість інформації, яка буде використовуватися при підтвердженні замовлення іншими операторами. Тому існує потреба перевірити функцію, що перевіряють коректність даних про бронювання. Розглянемо існуючі види запитів для бронювання:

- час та дата;
- кількість осіб;
- номер телефону;
- назва закладу;
- статус замовлення.

Створимо клас `BookingTest`, в якому будуть міститися тести для модуля бронювання. Перевірка буде здійснюватися за допомогою класу методів з фреймворку `unittest`. Успішність проходження тестів для процесу бронювання зображена на рисунку 5.1.



Unittests for test.BookingTest: 6 total, 6 passed		37 ms
		Collapse Expand
test		37 ms
BookingTest		37 ms
test_date_for_booking	passed	3 ms
test_insert_phone	passed	30 ms
test_invalid_amount_of_users	passed	1 ms
test_invalid_date_for_booking	passed	1 ms
test_invalid_phone_for_booking	passed	0 ms
test_valid_number_for_booking	passed	2 ms

Рисунок 5.1 – Звіт з проходження тестів для модулю бронювання

Пошук закладів є одною з найважливіших функцій системи. Потрібно перевірити чи спрацьовують функції, що перевіряють текстовий запит користувача на коректність. Наведемо перелік функцій, що потрібно перевірити при тестуванні у таблиці 5.13.

Таблиця 5.13 – Перелік функцій, що використовуються у процесі тестування.

Назва функції	Опис функції	Вхідні параметри
check_street	Перевіряє чи належить слово до регіону, вулиці, метро міста.	Текстовий запит
check_sklon_and_common_prefix	Перевіряє відмінок слова та повертає слово у називному відмінку.	Текстовий запит
check_synonyms	Перевіряє слово на наявність синонімів.	Текстовий запит, словник синонімів
check_len	Перевіряє слово на кількість символів.	Текстовий запит
check_form	Перевіряє чи слово у множині чи однині.	Текстовий запит
check_in_interval	Якщо декілька слів у запиті перевіряє наявність словосполучення у меню в заданому діапазоні.	Текстовий запит, інтервал
check_tags	Перевіряє чи слово відноситься до категорії.	Текстовий запит, словник категорій

Результати тестування даного модулю наведені на рисунку 4.2

TextRequestTest		121 ms
test_check_form	passed	15 ms
test_for_check_street	passed	10 ms
test_for_using_synonym	passed	2 ms
test_invalid_for_exception_invalid_symbol	passed	5 ms
test_invalid_for_multiplie_word	passed	70 ms
test_invalid_for_range_in_menu	passed	5 ms
test_invalid_for_russian	passed	4 ms
test_tags	passed	10 ms

Рисунок 5.2 – Звіт з проходження тестів для текстових запитів

На етапі розробки при написанні складних функцій виникає потреба протестувати проміжні частини коду. Наприклад, у випадку, що надсилають запити зі сторонніх сервісів не є реалізованими або неможливо підключитися до бази даних, бо вона ще не наповнена даними. У модульному тестуванні дуже часто використовуються мок-об'єкти, що вирішують дану проблему. Даний інструмент імітує роботу важливих частин коду. Фреймворк unittest підтримує дану технологію. Для того, щоб протестувати функції інтерфейсу чат-бота підключення до API-серверу замінимо запит, що повертає інформацію про заклад на мок-об'єкт. Приклад програмного коду наведений на рисунку 5.3.

```
class OffersTest(TestCase):
    @patch('main.Offers')
    def test_mock_for_offers(self, MockOffers):
        res = MockOffers()

        res.more_request_menu.return_value = [
            {
                {
                    "address": "вул. Велика Васильківська, 5, Арена Сіт",
                    "phone": "xxxxxx",
                    "text": "Ресторан спеціалізується на приготуванні б",
                    "text_id": "10100441",
                    "price": "",
                    "w_from": "00:00:00",
                    "place_name": "The Burger",
                    "w_to": "00:00:00",
                    "day": 1,
                    "expire_date": "",
                    "priority": 0,
                    "links_gm": "https://www.google.com.ua/maps/place/T",
                    "length": "0.15",
                    "lat": "50.439582",
                    "long": "30.520882",
                }
            }
        ]

        response = res.posts()
```

Рисунок 5.3 – Приклад мок-об'єкту для запиту

Тестування інтерфейсу

Для того, щоб інтерфейс в Telegram відображав правильну інформацію потрібно перевірити його взаємодію з API. Запити, що має даний інтерфейс наведені у другому розділі. Перевіримо коректність запитів за допомогою додатку PostMan, що надає зручний інтерфейс для тестування. На рисунках 4.5-4.8 зображені приклади тестів.

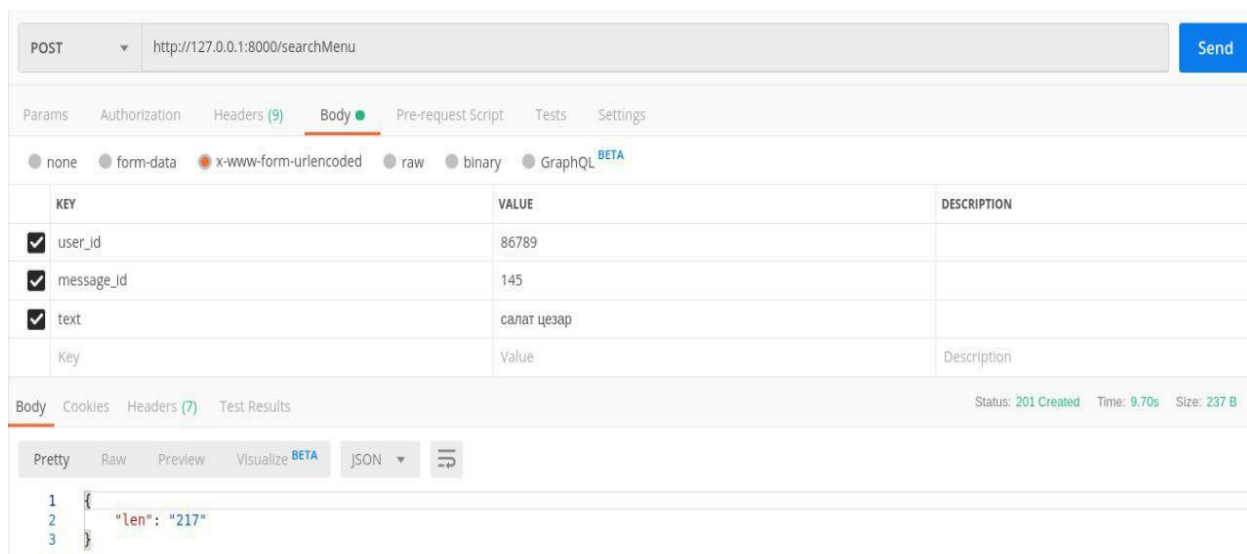


Рисунок 5.5 – Тестування запиту, що здійснює пошук текстового закладу

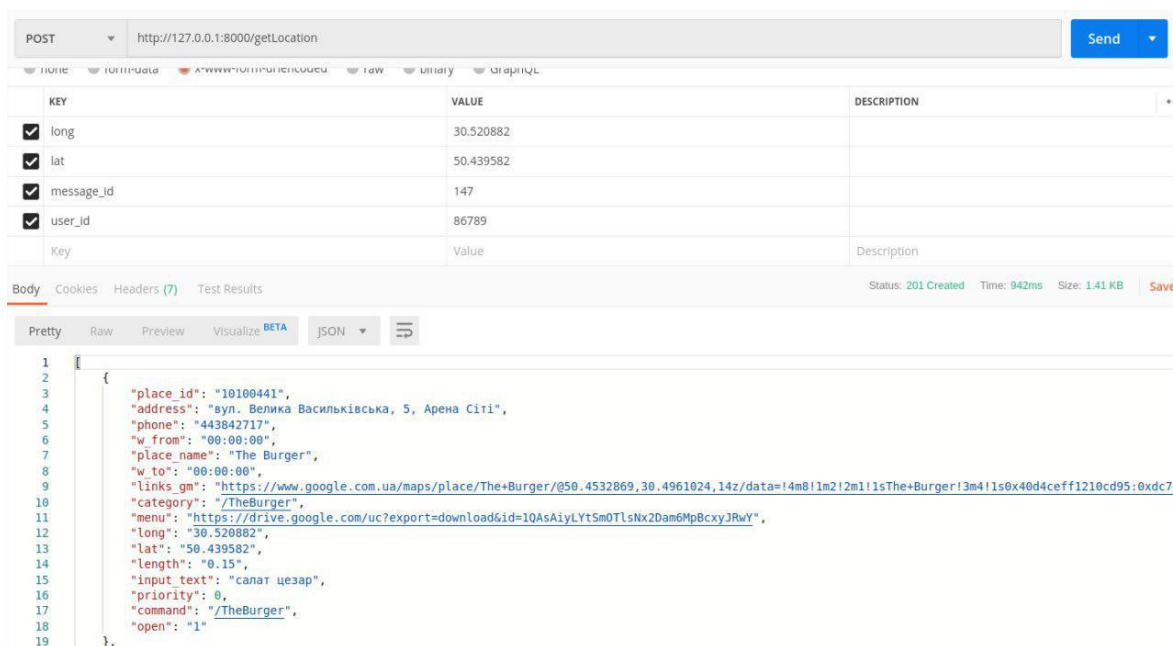


Рисунок 5.6 – Тестування запиту з місцезнаходженням

Висновок до розділу

У даному розділі були застосовані методи тестування системи, а саме ручне тестування та модульне тестування.

Для кожної з вимог системи був створений сценарій ручного тестування з описом результатів тестів. Модульне тестування було здійснено за допомогою фреймворку unittest. Даним методом був протестований процес бронювання столиків та пошуку закладів. Для перевірки роботи серверної частини був використаний додаток PostMan. Всі тести були пройдені успішно, що свідчить про розроблювана система працює правильно та задовольняє всі вимоги.

6 РОЗРОБЛЕННЯ СТАРТАП ПРОЕКТУ

6.1 Опис ідеї проекту

Щоб достовірно визначити можливості даного проекту конкурувати на ринку опишемо ідею програмного продукту (таблиця 6.1)

Цільова аудиторія проекту поділяється на дві групи:

- клієнти закладів харчування;
- власники закладів харчування.

Клієнти шукають пропозиції щодо закладів харчування відносно їх вподобань та місцезнаходження. Власники зацікавлені у інформації щодо клієнтських запитів та у відображенні їх закладу у програмному продукті, що містить актуальну інформацію про заклад та автоматизує процеси бронювання столиків.

Система буде користуватися попитом тому, що вона є потужним та гнучким інструментом пошуку закладів харчування та страв для користувачів. Метою проекту є заохочення потреб клієнтів та власників закладів харчування.

Таблиця 6.1 – Опис проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача

Отже, описавши ідею проекту визначено, що даний проект є перспективним та охоплює велику аудиторію користувачів.

Наступним етапом проведемо аналіз техніко-ідейних характеристик конкурентних продуктів. Потрібно порівняти наявність характеристик серед наявних конкурентів.

Таблиця 6.2 – Визначення характеристик ідеї проекту

Техніко-економічні характеристики ідеї	Конкуренти				W	N	S
	Мій проєкт	Sirved	Hungry GoWhere	OpenTable			
Пошук по меню			+	+		+	
Пошук по місцезнаходженню			+	-			+
Платформи, що підтримуються		-	-	+	+		
Можливість бронювання столиків		-	+	+		+	
Співробітництво з закладами			+	+		+	
Збір аналітичних даних			-	-			+

Проведемо аналіз технологій, які будуть основними в реалізації ідеї, де будуть ураховуватися такі фактори, як наявність на доступність технологій (таблиця 6.3)

Таблиця 6.3 – Технологічна здійсненність ідеї проекту

Ідея проекту	Технології реалізації	Наявність технологій	Доступність технологій
Розпізнавання меню	Оптичне розпізнавання тексту	Наявна	Доступна, але існує обмеження на знаки за відсутності платної підписки
Створення чат боту	Telegram API, клієнтська мова програмування	Наявна	Доступна
Адміністрування даних та модуль аналізу	Серверна мова програмування	Наявна	Доступна
Робота з даними	API	Наявна	Доступна

Технологічний аудит проекту дозволяє оцінити доступність реалізації ідеї. Можна зробити висновок, що всі вищенаведені технології, які будуть використані в проекті є доступними. Потреба у додаткових коштах з'являється у випадку, коли кількість даних для розпізнавання, перевищує кількісне обмеження.

6.2 Аналіз ринкових можливостей запуску стартап-проекту

Проаналізуємо потенційний ринок даного проекту, проаналізувавши кожен з наступних показників [29]. Кількість головних гравців проекту складає 3 одиниці, а загальний обсяг продаж 1000 грн за пакет послуг сервісом в місяць. Система немає обмежень для входу на ринок, але динаміка ринку зростає через наявність

конкурентів. Отже, не дивлячись на конкуренцію, ринок є привабливим для стартап-проекту.

Визначимо основні групи клієнтів на основі потреб, що формує ринок (таблиця 6.4).

Таблиця 6.4 – Опис потенційних клієнтів проекту

Потреба	Цільова аудиторія	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
Пошук місця, де можна похарчуватися згідно з вподобань та місцезнаходження	Жителі міста, туристи	Поведінку формують вподобання користувача: - місцезнаходження, - цінова категорія, - різноманіття пропозицій.	До продукції: швидкий та інтуїтивний спосіб пошуку та релевантність запропонованої інформації
Збільшення кількості клієнтів	Власники закладів харчування	Націленість на зростання попиту	Актуальність інформації та зворотній зв'язок
Статистика попиту	Маркетологи, таргетологи	-	Достовірність даних

Дуже важливо взяти увагу з якими загрозами може стикнутися проект в майбутньому. Розглянемо потенційні загрози, а саме актуальність інформації, зростання конкуренції, нереалістичні очікування.

Зростання конкуренції. Вищезгадані конкурентні проекти знаходяться не в нашій країні тому, існує загроза, що вони будуть масштабувати свій готовий сервіс.

Маючи гарну репутацію в інших країнах та пропонуючи власникам ресторанів гарні умови для співпраці, вони можуть знищити будь-який стартап-проект. Можливою реакцією компанії може бути удосконалення свого продукту та створення власної маркетингової стратегії націленої на ринок країни в якому проект буде впроваджений.

Нереалістичні очікування. Ринок кожної країни різний та сфера харчування має свої концепції. Складно спрогнозувати поведінку власників ресторанів тому й вимоги до цільових категорій різні. Компанії потрібно навчитися оперативно підлаштовуватися під ринок споживача, визначити та проаналізувати шаблони поведінки споживачів та тенденції в даній сфері.

Актуальність інформації. Процес налагодження зв'язків з закладами процес довготривалий, тому процес збору інформації з закладів є важким. Саме ці дані швидко втрачають свою актуальність. Наприклад, заклад може закритися або ціни в меню змінитися. Неактуальна інформація є загрозою та може підірвати довіру користувачів сервісу. Компанія, щоб уникнути ситуації повинна виділити кошти на інструменти для контролю за достовірністю інформації. Наприклад, можна наняти регіональних менеджерів, які будуть контролювати актуальність інформації на сайтах закладів. Крім, загроз існують фактори можливостей (таблиця 6.5).

Таблиця 6.5 – Фактори можливостей

Фактор	Зміст можливості	Можлива реакція компанії
Широка цільова аудиторія	Аудиторія продукту охоплює велику кількість людей, що сприяє простому розповсюдженню продукту	Розробити потужну рекламну кампанію
Пошук удосконалення процесу	Знаходження методів та технологій, що поліпшують процес пошуку	Пошук шляхів вдосконалення алгоритму пошуку продукту
Інтеграція з іншими	В сфері харчування існує багато сервісів, яким	Удосконалити архітектурні рішення для вдалої інтеграції

платформами	продукт може допомогти для розповсюдження своєї продукції	
Легке масштабування	Існує багато популярних платформ обміну миттєвими повідомленнями. Кожна платформа має різну аудиторію. Таким чином можна збільшити кількість користувачів.	Масштабувати інтерфейс на інші платформи

Одним з етапів аналізу є ступеневий аналіз конкуренції (таблиця 5.6) на ринку, що описує особливості середовища серед конкурентів та наслідки цих особливостей.

Таблиця 6.6 – Ступеневий аналіз конкуренції

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства
Монополія	Конкуренти такі як Sirved є унікальними в своїй країні.	Запропонувати користувачам продукт з кращими функціями, що будуть якісніше чим у конкурентів продукту.
Світова (за рівнем конкурентної боротьби)	Зараз програмно-конфігуруванні мережі активно впроваджують по всьому світу.	Спочатку потрібно отримати довіру на національному рівні для того щоб претендувати на світову конкуренцію
Внутрішньогалузева (за галузевою ознакою)	Внутрішньогалузева конкуренція тому, що	Знати всі аспекти та проблеми пов'язані з даною галуззю

	подібні продукти використовуються тільки в сфері закладів харчування.	
Товарно-видова (за видами товарів)	Ідея у конкурентних продуктів орієнтована на одну й ту ж потребу, але відрізняється функціями у мінімальному обсязі.	Слідкувати за конкуренцією аналогів та продуктів, зі схожими функціями
Нецінова (за характером конкурентних переваг)	Проект має цінність у функціях, якщо серед аналогів впроваджений якісний функціонал, то користувачі оберуть саме цей аналог.	Не впливає
Немарочна (за інтенсивністю)	Марка не впливає на продукт, впливає тільки якість запропонованих функцій.	Складніше рекламувати свій товар.

Проведемо аналіз конкуренції за Портером наведений в таблиці 6.7. Дана модель забезпечує ефективну стратегію для того, щоб виявити та проаналізувати фактори ризиків для проекту.

Таблиця 6.7– Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	HungryGoWhere	Мережеві закладів, що розроблюють для своїх закладів такі сервіси; сервіси, що пропонують акції та знижки	Необхідність постачальників відсутня	Жителі міст, туристи, власники закладів харчування, маркетологи	-
Висновки:	Прямі конкуренти існують, але на ринках інших країн	Все залежить від ефективності, адаптації та якості продукту	Постачальники не грають ролі в даному проекті	Неймовірна кількість клієнтів всіх видів, оскільки проблема є актуальною	Товари-замінники відсутні

Підсумовуючи вищенаведені висновки для кожного з аналізу, визначимо фактори конкурентоспроможності та обґрунтуємо їх значущість серед інших проектів.

Таблиця 6.8 – Обґрунтування факторів конкурентоспроможності

№	Фактор конкурентоспроможності	Обґрунтування
1	Швидкість пошуку	Користувачі очікують результатів пошуку якнайскоріше, якщо результати занадто довго обробляють це знижає зацікавленість до продукту
2	Точність розпізнавання меню	В випадку розпізнавання меню існують багато перешкод. Кожне індивідуальне меню має свої особливості. У випадку неправильного розпізнавання тексту результати будуть неточними. Тому важливо мати потужні інструменти для розпізнавання для меню.
3	Доступність інтерфейсу	Оскільки проблема пошуку закладів зазвичай виникає в дорозі, важливо мати інструмент, що доступний для кожного користувача на популярній платформі та швидко надсилає результат.
4	Актуальність наданої інформації	Якщо дані, що будуть надані користувачу продукту будуть неправильними це знизить репутацію продукту. Інформація повинна вчасно надаватися та бути актуальною.
5	Можливість бронювання	В деяких закладах харчування потрібно перевірити наявність вільних місць.
6	Універсальність продукту	Продукт може бути інтегрованим в інші сервіси, що пов'язані з доставкою

		продуктів. Важливо мати інструменти для інтеграції.
7	Аналітика для попиту	Вибір та вподобання користувачів впливають на якість сервісу. Аналіз таких даних призведе до збільшення прибутку користувачів

На основі факторів проведемо аналіз слабких та сильних сторін нашого продукту, що наведений у таблиці 6.9

Таблиця 6.9 – Порівняльний аналіз сильних та слабких сторін

№	Фактор конкурентоспроможності	Рейтинг товарів-конкурентів у порівнянні з системою						
		-3	-2	-1	0	+1	+2	+3
1	Швидкість пошуку			+				
2	Точність розпізнавання меню				+			
3	Актуальність наданої інформації				+			
4	Можливість бронювання					+		
	Універсальність продукту		+					
	Доступність інтерфейсу			+				
	Аналітика для попиту	+						

Для того, щоб розробити бізнес-стратегію для стартап-проекту використаємо аналіз, що приведений в таблиці 5.10. Це означає, що потрібно зробити висновки про речі, які можна контролювати та змінювати в проекті, а саме слабкі та сильні сторони

проекту та про те, що може вплинути на проект в майбутньому: можливості та загрози.

Таблиця 6.10 – SWOT- аналіз стартап-проекту

<u>Сильні сторони</u> <ul style="list-style-type: none"> - Доступність інтерфейсу - Пошуковий механізм 	<u>Слабкі сторони</u> <ul style="list-style-type: none"> - Обмеженість платформи - Контроль за зміною інформації
<u>Можливості</u> <ul style="list-style-type: none"> - Інтеграція - Масштабування - Розширення цільової аудиторії 	<u>Загрози</u> <ul style="list-style-type: none"> - Зростання конкуренції - Нереалістичні очікування - Неактуальність інформації

Таблиця 6.11 – Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Розроблення першої мінімальної версії продукту, збір відгуків від користувачів в одному місті	Висока	90 днів
2	Розроблення повноцінної версії, запуск по всій країні	Середня	6 місяці
3	Інтеграція з сервісами для доставки їжі, в якості пошукового механізму для залучення клієнтів	Середня	2 місяці

6.3 Розроблення ринкової стратегії проекту

В основу розроблення ринкової стратегії потрібно обрати стратегії серед таких фактори як розвиток, конкурентна поведінка та позиціонування продукту. В першу чергу визначимо цільові групи потенційних споживачів (таблиця 6.12).

Таблиця 6.12 – Вибір цільових груп потенційних споживачів

№	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Жителі міста, туристи	Висока	Високий попит	Середньої інтенсивності	Низька
2	Власники закладів харчування	Висока	Середній попит	Високої інтенсивності	Середня
3	Таргетологи	Середня	Середній попит	Низької інтенсивності	Проста
<p>Оскільки показники готовності споживачів сприйняття продукту та попиту є високими, то обираємо перший та другий сегмент споживачів.</p> <p>Отже, обираємо стратегію диференційованого маркетингу.</p>					

Для подальшої роботи з сегментами визначимо базову стратегію розвитку з конкурентних видів за М. Портером. Основою цих концепцій є те, що продукт повинен мати відмінну характеристику, що зробить його унікальним. Якщо цією особливості в продукті не буде, то він не споживач не буде зацікавлений в такому продукті. Вибираючи стратегію опишемо декілька ключових моментів щодо продукту:

- на ринку розроблюваний продукт є унікальним, але існують аналоги, що мають схожі функції та вже є конкурентоспроможними на ринку;
- компанія буде шукати нових споживачів, для просування проекту;

– для реалізації продукту компанія буде частково копіювати характеристики конкурентів, але за ціль ставиться виділяти ресурси на унікальний функціонал продукту.

Отже, найбільш доцільно вибрати стратегію диференціації, що пропонує споживачу продукт з унікальними споживчими характеристиками та підтримує цю концепцію довгостроково. Це дозволить в майбутньому призначати продукту більш високу ціну, збільшувати рівень прибутку та долучати більше споживачів.

Чим більш вузький сегмент, що займає продукт, тим легше спланувати його стратегію. Стратегія позиціонування[] потрібна для того, щоб визначити місце нового товару в свідомості у споживача

Таблиця 6.13 – Визначення стратегії позиціонування

Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту
Актуальність та доступність	Стратегія позиціонування категорії	Попит на новий підхід до вирішення проблеми.	Унікальний програмний продукт

6.4 Маркетингова програма стартап-проекту

Таблиця 6.14 – Визначення ключових переваг концепції потенційного товару

№	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами
1	Пошук місця, де можна похарчуватися згідно з вподобань та місцезнаходження	Програмний продукт, що автоматизовано надає послугу підбору закладів в зручному вигляді.	- зручний та доступний інтерфейс; - актуальність запропонованої інформації; - пошук по позиціям з меню.
2	Збільшення кількості клієнтів	Інтеграція з продуктом; Замовлення реклами та послуги бронювання столику.	можливість бронювання; вигідні умови підключення до продукту; замовлення їжі .
3	Статистика попиту	Перегляд статистики попиту та користування, інформація по кожному с закладів харчування, що може бути використана як таргетингу.	- формування звітів для кожного з клієнтів

Розробимо трирівневу маркетингову модель товару, опис моделі наведений с таблиці 6.18.

Таблиця 6.1. – Опис трьох рівнів

Рівні	Сутність та складові	
Товар за задумом	Вигоди від використання програмного продукту “Рекомендаційна система пошуку закладів харчування на основі платформи обміну миттєвими повідомленнями”: <ul style="list-style-type: none">- швидкий пошуковий інструмент для підбору закладів, що економить час;- інструмент, що збільшує кількість відвідувачів (бронювання столиків, пошук);- відстеження інформації для SEO-оптимізації.	
	Властивості/характеристики	Монотонність
	1. Швидкість пошуку	М
	2. Актуальність інформації	Нм
	3. Якість розпізнаванню	Нм
	4. Кросплатформеність	М
	5. Масштабованість	М
	6. Зручність користування	
Якість: відповідність функцій, що включає продукт буде перевірятись здійсненням тестів, актуальність інформації буде контролюватися відповідними спеціалістами; технічна підтримка здійснюється технічними спеціалістами.		
Рекомендаційна система пошуку закладів харчування		
	До продажу: пошукова система, можливість бронювання, пакети послуг для власників ресторанів.	
	Після продажу: можливість пре замовляти товари, перехід на інші платформи, знижки.	

Висновок до розділу

У даному розділі був роз ідея стартап проекту на основі програмного забезпечення.

Був проведений аналіз ринкових можливостей проекту та створена маркетингова та ринкова стратегія просування системи. Були визначенні такі загрози, як конкуренція та неактуальність інформації.

Проект має перспективи на подальший розвиток та має багато переваг серед аналогів.

ЗАГАЛЬНИЙ ВИСНОВОК

У ході виконання роботи була створена рекомендаційна система пошуку закладів харчування на основі платформи обміну миттєвими повідомленнями.

Дана система спрощує процес підбору закладів харчування за рахунок автоматизації процесу пошуку та бронювання.

У першому розділі аналізу наявних аналогів системи були розглянуті та проаналізовані аналоги даної системи серед програмних продуктів, види рекомендаційних систем та платформ обміну миттєвими повідомленнями.

У другому розділі були наведені функціональні вимоги до системи, де описана та створена діаграма варіантів використання, та до кожної вимоги був наведений опис у вигляді таблиць.

Математичне забезпечення системи визначено у третьому розділі, в якому було обрано метод надання рекомендацій та метод розрахунку відстаней між користувачем та точкою на карті.

У розділі архітектури та проектування системи розглядаються основні процеси створення програмного продукту. Наведений опис структури бази даних, попередньої обробки текстових даних та архітектури програмного забезпечення.

Для перевірки коректності роботи розробленої системи був наведений розділ з тестуванням, що включає ручне та модульне тестування. Всі тести були пройдені успішно.

У розділі розроблення стартап проекту було розроблено ринкову та маркетингову стратегію системи та проведено аналіз конкурентів системи.

Розроблений програмний продукт був впроваджений у роботу підприємства. У майбутньому система може бути покращена за рахунок оптимізації методу розпізнавання меню та додавання модулю доставки їжі з можливістю оплати у системі.

ПЕРЕЛІК ПОСИЛАНЬ

1. “Chatbot Report 2019: Global Trends and Analysis” [Електронний ресурс] // Режим доступу: <https://digitalgo.it/en/blog/chatbot-future-instant-messaging/>
2. Fantoni, Beatrice; October 27, Windsor Star Updated Sirved [Електронний ресурс] // Режим доступу: <https://windsorstar.com/life/food/made-in-windsor-takeout-app-delivers-Gade-in-windsor-food>
3. Google Maps API [Електронний ресурс] // Режим доступу: https://cloud.google.com/maps-platform/?_ga=2.60205357.1424803530.1576574588-637138501.1576022847
4. HungryWhereToGo [Електронний ресурс] // Режим доступу: <https://www.hungrygowhere.com>
5. OpenTable, [Електронний ресурс] // Режим доступу: <https://press.opentable.com/news-releases/news-release-details/opentable-acquire-foodspotting?ReleaseID=73,295>
6. Ketakee Nimavat, Prof. Tushar Champaneria, «Chatbots: An overview Types, Architecture, Tools and Future Possibilities», pp.34-37
7. Architecture for public service chatbots [Електронний ресурс] // Режим доступу: https://joinup.ec.europa.eu/sites/default/files/news/2019-09/ISA2_Architecture%20for%20public%20service%20chatbots.pdf
8. Telegram Bot API Documentation [Електронний ресурс] // Режим доступу: <https://core.telegram.org/bots/api>
9. Developer Solution for a platform [Електронний ресурс] // Режим доступу: <https://developers.viber.com>
10. Facebook for developers [Електронний ресурс] // Режим доступу: <https://developers.facebook.com/docs/messenger-platform/>
11. D. Britz, “Deep Learning for chatbots.” [Online]. Available: www.wildml.com/2016/04/deep-learningfor-chatbots-part-1-introduction/
12. The Register, “Please don’t call them Facebook chatbots, says Facebook’s bot boss • The Register.” [Online]. Available: <https://www.theregister.co.uk>

13. Frias-Martinez, E., Magoulas, G., Chen, S. Y., and Macredie, R. "Automated user modeling for personalized digital libraries", International Journal of Information Management, , pp.234–248
14. Francesco Ricci, Lior Rokach, Bracha Shapira, Paul B. Kantor "Recommender Systems Handbook"
15. Ruzhi Xu et al., "Distributed collaborative filtering with singular ratings for large scale recommendation", Elsevier The Journal of Systems and Software
16. Jure Leskovec, Anand Rajaraman "Mining of Massive Datasets"
17. Robin Bruke "Recommender System, Overview" [Электронный ресурс] // Режим доступа: https://www.researchgate.net/publication/220604600_Recommender_Systems_An_Overview
18. Dietmar Jannach, Markus Zanker "Recommender Systems", p.45-47
19. Nitin Pradeep Kumar, Zhenzhen , "Hybrid User-Item Based Collaborative Filtering" 19th International Conference on Knowledge Based and Intelligent Information and Engineering Systems p.244-245
20. S.J. Gong, H.W. Ye, X.M. Zhu. Item-Based Collaborative Filtering Recommendation using Self-Organizing Map. Proceedings of Chinese Control and Decision Conference. 2009
21. S.J. Gong, H.W. Ye, X.M. Zhu. Item-Based Collaborative Filtering Recommendation using Self-Organizing Map. Proceedings of Chinese Control and Decision Conference. 2009
22. Python [Электронный ресурс] // Режим доступа: <https://docs.python.org/3/>
23. Flask [Электронный ресурс] // Режим доступа: <http://flask.pocoo.org/docs/0.12>
24. Django [Электронный ресурс] // Режим доступа: <https://www.djangoproject.com>
25. PostgreSQL [Электронный ресурс] // Режим доступа: <https://postgrespro.ru/docs/postgresql/9.6/>
26. GoogleOCR [Электронный ресурс] // Режим доступа: <https://console.developers.google.com/apis/dashboard?pli=1>
27. Nginx [Электронный ресурс] // Режим доступа: <https://nginx.org/ru/>